

# **Advantage™ VISION:Workbench™ for ISPF**

## **Reference Guide**

**6.0**



Computer Associates™

WKHUG060.PDF/D50-101-011

This documentation and related computer software program (hereinafter referred to as the "Documentation") is for the end user's informational purposes only and is subject to change or withdrawal by Computer Associates International, Inc. ("CA") at any time.

This documentation may not be copied, transferred, reproduced, disclosed or duplicated, in whole or in part, without the prior written consent of CA. This documentation is proprietary information of CA and protected by the copyright laws of the United States and international treaties.

Notwithstanding the foregoing, the user may print a reasonable number of copies of this documentation for its own internal use, provided that all CA copyright notices and legends are affixed to each reproduced copy. Only authorized employees, consultants, or agents of the user who are bound by the confidentiality provisions of the license for the software of the user will have access to such copies.

This right to print copies is limited to the period during which the license for the product remains in full force and effect. Should the license terminate for any reason, it shall be the user's responsibility to return to CA the reproduced copies or to certify to CA that same have been destroyed.

To the extent permitted by applicable law, CA provides this documentation "as is" without warranty of any kind, including without limitation, any implied warranties of merchantability, fitness for a particular purpose or noninfringement. In no event will CA be liable to the end user or any third party for any loss or damage, direct or indirect, from the use of this documentation, including without limitation, lost profits, business interruption, goodwill, or lost data, even if CA is expressly advised of such loss or damage.

The use of any product referenced in this documentation and this documentation is governed by the end user's applicable license agreement.

The manufacturer of this documentation is Computer Associates International, Inc.

Provided with "Restricted Rights" as set forth in 48 C.F.R. Section 12.212, 48 C.F.R. Sections 52.227-19(c)(1) and (2) or DFARS Section 252.227-7013(c)(1)(ii) or applicable successor provisions.

© 2002 Computer Associates International, Inc. (CA).

All trademarks, trade names, service marks, and logos referenced herein belong to their respective companies.

# Contents

## Chapter 1: Introduction

Overview of VISION:Workbench for ISPF .....	1-1
Creating a VISION:Transact Application .....	1-3
Online Application Summary .....	1-7
Creating a VISION:Builder Application .....	1-7
Batch Application Summary .....	1-11
Creating a File Definition .....	1-11
Using the VISION:Workbench for ISPF Utilities .....	1-14
Using the Help Facility .....	1-17

## Chapter 2: Using VISION:Workbench for ISPF

Similarities to ISPF .....	2-1
VISION:Workbench for ISPF Screen Formats .....	2-4
Panel Name .....	2-5
Commands .....	2-8
Primary Commands .....	2-9
Line Commands .....	2-13
Interactive Help Facilities .....	2-16
Split Screen Processing .....	2-20
Editing Items .....	2-21
Step <a href="#">1</a> : Select an Item Type .....	2-22
Step <a href="#">2</a> : Specify Item Location .....	2-22
Step <a href="#">3</a> : Specify Validation Libraries (VISION:Builder only) .....	2-30
Step <a href="#">4</a> : Edit the Selected Item .....	2-31
Step <a href="#">5</a> : Validate the Item .....	2-32
Step <a href="#">6</a> : Saving the Item .....	2-37

---

Introduction to the Import Option .....	2-38
COBOL Definitions .....	2-40
DB2 Table Definitions .....	2-46
VISION:Results Definitions .....	2-50
VISION:Inquiry Definitions.....	2-53
VISION:Builder Definitions.....	2-56
Duplicate Field Name Check.....	2-57
Duplicate Field Name Editor.....	2-59
Introduction to Job Submission.....	2-60
How the Submission Facility Works .....	2-60
Using the Submission Facility .....	2-62
Step 1: Specify Item to Submit.....	2-63
Step 2: Supplying File Information.....	2-64
Step 3: Completing the Options Panel .....	2-66
Step 4: Completing the User Panel .....	2-67
Step 5: Generating JCL or CLISTS .....	2-68
Using the Utilities Subsystem.....	2-68
VISION:Workbench for ISPF Parameters Subsystem .....	2-69
User Profile - First Time User Setup .....	2-70
Setting Session Parameters .....	2-71
Setting List Data Set Parameters .....	2-75
Setting Item Entry and Validation Library Defaults .....	2-76
Setting the Screen Developer Profile .....	2-76
Setting Default Job Submission Information.....	2-77
Allocating ISPF FILE .....	2-82
Allocating ISPF FILE for VISION:Transact.....	2-83
Allocating ISPF FILE for VISION:Builder.....	2-83
File Tailoring Output Data Set Characteristics.....	2-84
The List Data Set.....	2-85
Using Navigation Short Cuts .....	2-87
Using Command Stacking .....	2-87
Using the Option Jump Function .....	2-91
Using Multiple Line Command Selections.....	2-94
Maintaining Existing Source.....	2-96
VISION:Builder and VISION:Transact Source Importing Guidelines .....	2-96
VISION:Transact Conversions .....	2-97
Converting Run Control Information .....	2-97
Converting Screen Control Statements .....	2-98
VISION:Builder Validation Considerations .....	2-100
DB2 Considerations .....	2-100
Sample Reports.....	2-101
VISION:Transact Validation Considerations .....	2-103
DB2 Considerations .....	2-104
Catalog Considerations .....	2-104

---

Using the Screen Developer.....	2-105
Completing Screen Developer Options .....	2-105
Using Special Screen Developer Primary Commands.....	2-124
ASL Procedure Validation.....	2-127
Using Comment and CALL Statements.....	2-128
The Unexpected Error Panel.....	2-129
Contacting Computer Associates.....	2-130

## Index



# Introduction

---

This chapter provides you with a brief introduction to VISION:Workbench™ for ISPF and gives you an overview of the product by describing some examples. It illustrates only those panels most relevant to the example. This chapter does not attempt to teach you all of the codes and commands that are available when using VISION:Workbench for ISPF. [Chapter 2, Using VISION:Workbench for ISPF](#) of this guide introduces you to these. The extensive Help system of over 700 panels is also discussed in [2](#).

This chapter covers the following:

- [Overview of VISION:Workbench for ISPF on page 1-1](#)
- [Creating a VISION:Transact Application on page 1-3](#)
- [Creating a VISION:Builder Application on page 1-7](#)
- [Creating a File Definition on page 1-11](#)
- [Using the VISION:Workbench for ISPF Utilities on page 1-14](#)
- [Using the Help Facility on page 1-17](#)

## Overview of VISION:Workbench for ISPF

VISION:Workbench for ISPF is an interactive development facility for creating and maintaining your VISION:Transact™ (online) and VISION:Builder® (batch) applications and definitions. In this friendly interactive environment, you can develop applications quickly and easily. VISION:Workbench for ISPF also has a facility for creating and maintaining VISION:Inform® definitions.

VISION:Workbench for ISPF allows you to describe your applications on a series of panels. By filling in appropriate entries on these panels, an entire application can be described in a matter of minutes. The same VISION:Workbench for ISPF panels you use to create new applications and definitions can also be used to maintain any existing VISION:Builder and VISION:Transact applications or definitions.

In this interactive environment, individual entries are syntax checked and validated as you enter them. If VISION:Workbench for ISPF finds an incorrect entry, it displays an informational message to help you provide the correct information. If you do not know what to enter, there is an extensive Help facility that can provide detailed information on how to complete the entry. The information you need is always online.

For special assistance, an in-depth tutorial is provided. It can be used by new users who want to become familiar with the system or by experienced users seeking additional information on specific topics.

VISION:Workbench operates within the increasingly popular and friendly ISPF environment. It takes full advantage of the following ISPF facilities:

- Standard ISPF primary and line commands.
- Industry-wide standard panel formats and PF key usage.
- Menu selections and pre-determined paths to application building and maintenance.
- Navigational short cuts between panels.
- An extensive online Help facility at both the panel level and the field level.
- An in-depth tutorial.

In addition to creating complete applications, you can also use VISION:Workbench for ISPF to create individual data definitions, such as file definitions. VISION:Workbench for ISPF lets you define all types of file structures from simple flat files to complex relational databases. The definition process also extends to tables, transactions, reports, and the procedural requests that tie entire applications together.

With the same ease used to create file definitions, you can use VISION:Workbench for ISPF to paint VISION:Transact application screens, too.

VISION:Workbench for ISPF provides an additional feature to VISION:Transact application developers – the ability to preview how the screen looks and acts. You will not have to do a lot of compiles to find out your screen does not work the way you expected or that the screen field you want highlighted is blinking instead.

When creating individual data definitions, each definition is saved as a separate item. These individual components can be reused whenever and wherever you like. Create entirely new applications using components already used in other applications. There is no limit to the number of times you can reuse a component. The development cost savings realized from this feature alone makes VISION:Workbench for ISPF the most important application development tool your installation can use.

Once your application is error free and ready for processing, submit it to TSO directly from your VISION:Workbench for ISPF session. VISION:Workbench for ISPF contains panels that provide all the information for foreground or background processing. You can even view the results online.



VISION:Workbench is easily installed and modified to meet your particular installation standards. Because it is like ISPF, its panels can be tailored to meet your unique requirements. PF key designations can be changed. Your unique JCL and processing requirements can be customized.

Each of the examples use the Employee file, EMPFILE. The record is shown in [Figure 1-1](#).

EMPNUM	EMPNAME	EMPTITLE	EMPSAL	EMPSEX	EMPDIV	EMPFILL
--------	---------	----------	--------	--------	--------	---------

Figure 1-1 The Employee File - Record Definition

The Employee file is a simple flat file. VISION:Workbench for ISPF lets you define extremely complex file structures and interrelationships. However, for the sake of clarity, the examples in this guide were kept simple. Complex definitions would only interfere with the purpose of this guide, which is to illustrate how to use VISION:Workbench for ISPF.

**Note:** This guide also intentionally leaves out initial and intermediate panels to quickly illustrate the important ones.

## Creating a VISION:Transact Application

In this example, you create a VISION:Transact online application that uses the data entry screens shown in [Figure 1-2](#).

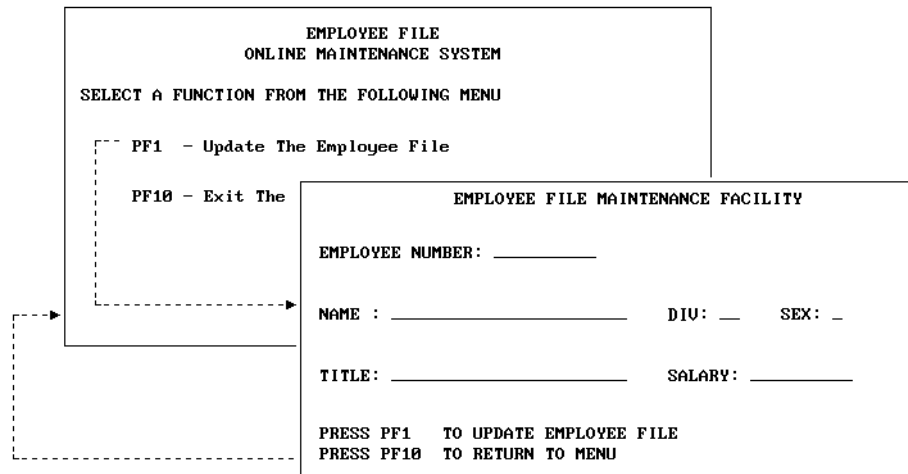


Figure 1-2 Sample Online Application Data Entry Screens

The following lists the steps involved in the development of this VISION:Transact application.

- 1. Identify the data file involved.
- 2. Create the two data entry screens shown in [Figure 1-2](#).
- 3. Define the database updates.
- 4. Link the screens together.

Step 1 is to identify the file. Identifying the file is done from the VISION:Workbench for ISPF Application Files panel shown in [Figure 1-3](#).

APP5FIL ----- MARKISPF.ISPLAG1.SOURCE(EMUPDATE) ----- ROW 1 OF 1  
COMMAND ==> SCROLL ==> CSR

APPLICATION FILES

Line	File	File	Synchronizing		Psd	Syn	Strg	Path	Recl	Creator	Ins	
Cmd	Type	Name	Qlf	Field 1	Field 2	File	Typ	Type	Call	Field	ID	Def
'''	FILE	EMPFILE	-			-	U	-	-			-
'''			-			-		-	-			-
'''			-			-		-	-			-
'''			-			-		-	-			-
'''			-			-		-	-			-
'''			-			-		-	-			-
'''			-			-		-	-			-
'''			-			-		-	-			-
'''			-			-		-	-			-
'''			-			-		-	-			-

\*\*\*\*\* BOTTOM OF DATA \*\*\*\*\*

Figure 1-3 Identifying the Data File to be Processed

To identify the file, specify the name of the file (EMPFILE). For this application, also specify that the file can be updated. Step 1 is complete.

Step 2 is the creation of the data entry screens. This is done by painting the screens exactly as you want them to appear.

In the following Screen Layout (SCRNLAYO) panels, the first three lines of the screen are VISION:Workbench for ISPF; the remaining lines are used to paint the screen.

[Figure 1-4](#) shows the creation of the initial menu screen and [Figure 1-5](#) shows the creation of the actual file maintenance screen.

```

SCRNLAY0 ----- MARKISPF.ISPLAG1.SOURCE(EMUPDATE) -----
COMMAND ==>
Line
Cmd
      EMPLOYEE FILE
      ONLINE MAINTENANCE SYSTEM
      SELECT A FUNCTION FROM THE FOLLOWING MENU:
      PF1  - Update The Employee File
      PF10 - Exit The Employee File Maintenance System

```

Figure 1-4 Painting the Initial Menu Screen

```

SCRNLAY0 ----- MARKISPF.ISPLAG1.SOURCE(EMUPDATE) -----
COMMAND ==>
Line
Cmd
      EMPLOYEE FILE MAINTENANCE FACILITY
      EMPLOYEE NUMBER: _____
      NAME : _____ DIV: __ SEX: _
      TITLE: _____ SALARY: _____
      PRESS PF1  TO UPDATE EMPLOYEE FILE
      PRESS PF10 TO RETURN TO MENU

```

Figure 1-5 Painting the File Maintenance Screen

Notice how the screen layouts have been drawn exactly as the screen is to appear in the application.

Painting provides you with the layout and tells you what the screen is going to look like. It does not tell you where to get the data to display on the screen or where to put the data that is entered on the screen. So next you need to map the data to and from the screen. This is achieved by using the panel shown in [Figure 1-6](#) to define the screen fields.

SCRNFlds ----- MARKISPF . ISPLAG1 . SOURCE(EMUPDATE) -----

COMMAND ==>

SCROLL ==> CSR

EMPLOYEE FILE MAINTENANCE FACILITY

EMPLOYEE NUMBER: \_\_\_\_\_

Line	In/	Screen	Q Dest	A	Q Source	Screen	Virtual	Layout
Cnd	Out	Field	L Field	C	L Field	Segment	Field	Position
		Name	F Name	T	F Name	Number	? Len	Line Column
0001	B	EMPNUM	_ EMPNUM	U	_ EMPNUM	01	_ _	7 18
0002	B	EMPNAME	_ EMPNAME	R	_ EMPNAME	01	_ _	10 8
0003	B	EMPDIV	_ EMPDIV	R	_ EMPDIV	01	_ _	10 38
0004	B	EMPSEX	_ EMPSEX	R	_ EMPSEX	01	_ _	10 50
0005	B	EMPTITLE	_ EMPTITLE	R	_ EMPTITLE	01	_ _	13 8
0006	B	EMPSAL	_ EMPSAL	R	_ EMPSAL	01	_ _	13 41
0007	-	_____	_____	N	_____	_____	_____	_____
0008	-	_____	_____	N	_____	_____	_____	_____

Figure 1-6 Identifying the Screen Data Fields

Step [3](#) is the definition of screen fields. This step allows you to:

- Identify from where the screen data is to come.
- Identify to where the screen data is to go.

For your convenience, the top part of the Screen Field panel enables you to look at the layout of the screen as you identify the purpose of each screen field.

Step [4](#) completes this application by linking the screens together. This is achieved by using the Screen and Application Flow panel shown in [Figure 1-7](#).

APP5FLOW ----- MARKISPF.ISPLAG1.SOURCE(EMUPDATE) -----										Row 1 of 5	
COMMAND ==>										SCROLL ==> CSR	
										SCREEN: EMMENU	
SCREEN AND APPLICATION FLOW											
Line	L	C	Q	Operand	O	Q	Operand	Output	Q	Operand	Partial
	U	O	L	...A....	P	L	...B....	Screen	L	...C....	S N O
Cnd	L	N	F	Name	R	F	Name/Constant	Name	F	Name	T O P
''''	-	-	F	ATTENID	EQ	C	PF1		-	-	-
''''	-	-	-	_____	OS	-	_____	EMBROWSE	-	-	-
''''	-	O	F	ATTENID	EQ	C	PF10		-	-	-
''''	-	-	-	_____	OS	-	_____	QUIT	-	-	-
''''	-	E	-	_____	OS	-	_____	EMMENU	-	-	-
''''	-	-	-	_____	-	-	_____		-	-	-
''''	-	-	-	_____	-	-	_____		-	-	-
''''	-	-	-	_____	-	-	_____		-	-	-
''''	-	-	-	_____	-	-	_____		-	-	-
''''	-	-	-	_____	-	-	_____		-	-	-

Figure 1-7 Linking the Screens

The Screen and Application Flow panel states, using pseudo code, the following when the input screen is EMMENU:

- If the attention key is PF1, the output screen is EMBROWSE.
- Or, if the attention key is PF10, the output screen is QUIT.
- ELSE, the output screen is EMMENU.

The application is complete.

## Online Application Summary

This section has shown you how complete VISION:Transact online application development is achieved with VISION:Workbench for ISPF. The VISION:Workbench for ISPF system presents panels that list the various components of the application. You simply select the pieces that you want to complete.

The same simplicity applies to all VISION:Transact and VISION:Workbench for ISPF capabilities. VISION:Workbench for ISPF provides complete flexibility for inputting, editing, constructing, and understanding your application.

VISION:Transact provides the capabilities for generating applications for IMS or CICS/MVS®, and for integrating complex databases such as IMS/DB or DB2®. In addition, VISION:Transact permits joins between IMS/DB, DB2, and VSAM. Complex screen interactions can be developed with extensive validation and automatic transaction back out in the event of a data error.

VISION:Workbench for ISPF is the easiest way to generate online applications.

VISION:Transact is the easiest way to execute online applications.

## Creating a VISION:Builder Application

In this example, you will create a VISION:Builder batch application using VISION:Workbench for ISPF. Once the application is created, VISION:Workbench for ISPF can also be used to automatically generate JCL and submit the application.

Keep in mind, VISION:Builder can handle extremely complex data structures and synchronize many different databases at one time. It can apply transactions and update an old master file to create a new master file. It can create as many as 255 reports in one pass of the database and output up to 10 subfiles. It can join up to 10 different databases. On top of this, it has a powerful procedural language for coding complex algorithms.

In this example, you will create a VISION:Builder application that will generate a report from the Employee file. This application will select all female employees who make at least \$75,000 a year. In addition, for the selected group, the report will display the following:

- The number of females meeting this criteria.
- The cumulative salary for the entire report.
- The highest salary.
- The lowest salary.
- The average salary.

The report that you want to generate is illustrated in [Figure 1-8](#).

FEMALE EMPLOYEES MAKING AT LEAST \$75,000				PAGE 1
EMPLOYEE NUMBER	EMPLOYEE NAME	EMPLOYEE TITLE	SALARY	
00000015	BANNER, NANCY	PURCHASING AGENT	79,000	
00000030	BOOTHE, TERESA	PRESS LIASON	96,000	
00000026	CARBONE, SANDRA	ADMIN ASSISTANT	90,340	
00000001	CARMELLO, SANDRA	MARKETING DIRECTOR	75,500	
00000020	DE DAVENTRY, ROSELLA	HEIR APPARENT	100,000	
00000037	GENET, JEAN	SPECIAL CONSULTANT	90,000	
00000005	HORSLEY, MARGARET	EDUCATION DIRECTOR	80,000	
00000027	KRAUS, ARTHEA	APPRAISER	81,000	
00000033	LAFABRE, GENESTA	PROTOCOL CHIEF	101,000	
00000036	LAFFLER, EVE	NEVADA BUREAU CHIEF	80,250	
00000029	LUNIER, FAITH	ADMINISTRATOR	105,000	
00000035	MALCOEUR, LALOTTE	ETHICS ENGINEER	92,000	
00000034	MORSE, MELANIE	ASST TO THE CEO	77,500	
00000024	NORMAN, ELIZABETH	DIRECTOR OF FINANCE	83,300	
00000014	RABIN, JESSICA	STAFF NURSE	92,000	
00000039	SANCHEZ, LISA	CITY HALL REP	88,140	
00000028	STARLIGHT, JOAN	PURCHASING AGENT	78,000	
00000016	STEIN, LISA	LEGAL ADVISOR	83,000	
00000012	VAN STINE, BARBARA	CEO	265,000	
GRAND	CUM.		1,837,030	
	COUNT		19	
	MAX.		265,000	
	MIN.		75,500	
	AVG.		96,685	

Figure 1-8 The 75K+ Report

To create this application, you need to take the following steps:

1. Identify the data file involved.
2. Specify the selection criteria.
3. Define a report.

Step [1](#) is to identify the files involved. This is illustrated in the Application Files panel shown in [Figure 1-9](#).

APP4FIL ----- MARKISPF.ISPLAG1.SOURCE(EMREPT) ----- ROW 1 OF 2  
COMMAND ==> SCROLL ==> CSR

APPLICATION FILES

Line Cmd	File Type	File Name	Array ? Equ	Mos ?	SQL/ DB2?	Cord Type	Key Qlf	Alt Key	Instr Def ?	Override DD/DTF
'''	M4OLD	EMPFILE	-	-	-	-	-	_____	-	_____
'''	M4REPO	_____	-	-	-	-	-	_____	-	_____
'''	_____	_____	-	-	-	-	-	_____	-	_____
'''	_____	_____	-	-	-	-	-	_____	-	_____
'''	_____	_____	-	-	-	-	-	_____	-	_____
'''	_____	_____	-	-	-	-	-	_____	-	_____
'''	_____	_____	-	-	-	-	-	_____	-	_____
'''	_____	_____	-	-	-	-	-	_____	-	_____
'''	_____	_____	-	-	-	-	-	_____	-	_____
'''	_____	_____	-	-	-	-	-	_____	-	_____

Figure 1-9 Specifying Application Files

Here you identify the old master file with its ddname of M4OLD and the file definition name of EMPFILE. You also specify that a work file, M4REPO, will be used to hold the reports.

Step 2. specifies the selection criteria as shown in [Figure 1-10](#).

ASL4 ----- ISPLCB1.MISPF.SOURCE(SALREPT) ----- ROW 1 OF 12  
COMMAND ==> SCROLL ==> DATA  
PROCEDURE: SALREPT

Procedure Type ==> \_ Max Items ==> \_\_\_\_\_  
Reinit Temps? ==> \_

Line Free Form Processing Statements  
Cmd -----

''' IF EMPSEX = 'F' AND EMPSAL >= 75000 THEN  
''' CALL REPORT RPTS  
''' ELSE  
''' CONTINUE  
''' END  
'''  
'''  
'''  
'''  
'''  
'''  
'''  
'''  
'''  
'''  
'''  
\*\*\*\*\* BOTTOM OF DATA \*\*\*\*\*

Figure 1-10 Specifying the Selection Criteria

Note that you can code the whole expression in one simple statement using the free-form ASL language (Advanced Syntax Language). (For more information about ASL, please see the ASL Reference Guide.)

Step 3 is to define the report in the Output Specifications panel as shown in Figure 1-11.

REPTFLDS----- MARKISPF . ISPLAG1 . SOURCE (EMREPT) -----										ROW 1 OF 4				
COMMAND ==>										SCROLL ==> CSR				
OUTPUT SPECIFICATIONS										REQUEST: RPTS				
										OUTPUT : 1				
										----- Summaries -----				
										Part1				
Line	Sp	L	Field	E N	Sort	Break				Tot	Cnt	Max	Rat	Field Output
Cmd	Seq	Bf	F	Name	D R	Seq/Des	Ctl/Sub			Cum	Min	Avg	Pct	ST NO Edit?
'''				EMPNUM										
'''				EMPNAME		1								
'''				EMPTITLE										
'''				EMPSAL						G	G	G	G	
'''														
'''														
'''														
'''														
'''														
'''														
'''														

Figure 1-11 Specifying the Contents of a Report

To define the report, list the names of the fields to be printed. You want the report sorted in alphabetical order by employee name. The sort sequence is identified by making EMPNAME the level one sort sequence. You also want to count the employees and obtain the cumulative, maximum, minimum, and average salaries for these employees. This is done by entering G under the appropriate columns.

Other than typing the field names, the entire report is described in six entries. Of course, no report should be without a title; a title is defined on the Report Title panel shown in Figure 1-12.

REPTTITL----- MARKISPF . ISPLAG1 . SOURCE (EMREPT) -----										ROW 1 OF 1				
COMMAND ==>										SCROLL ==> CSR				
REPORT TITLE										REQUEST: RPTS				
										OUTPUT : 1				
Line	Seq		Text											
Cmd	Num		....x....1....x....2....x....3....x....4....x....5....x....											
'''			FEMALE EMPLOYEES MAKING AT LEAST \$75,000											
'''														
'''														
'''														
'''														
'''														
'''														

Figure 1-12 Specifying the Report Title



## Batch Application Summary

To show you the simplicity of VISION:Workbench for ISPF, the application was kept simple. But, there are many additional things you could have done to implement an extremely complex application.

The fourth generation language (ASL) itself is extremely powerful and can be used to implement complex algorithms over complex databases. It has the arithmetic and logical expressions that you would expect to find in any 4GL language. It also has powerful text manipulation capabilities along with a rich variety of powerful operators.

There are many capabilities that can be used to design reports literally the way that you would like to see them. While you have seen how to create a single report, you could have developed 255 reports with the same simplicity as the one you just created. And all this applies to joins of IMS, DB2, VSAM, and other databases.

VISION:Workbench for ISPF is the easiest way to generate batch applications.

VISION:Builder is the easiest way to execute batch applications.

## Creating a File Definition

VISION:Workbench for ISPF provides a powerful facility for creating file definitions. VISION:Builder, VISION:Inform, and VISION:Transact can support the processing of complex hierarchical files or interrelated DB2 tables composed of an extensive number of fields with different formats. However, because the purpose of this document is to help you understand VISION:Workbench for ISPF, the example will use the simple Employee file to demonstrate the VISION:Workbench for ISPF file definition process.

As with the previous examples, some panels are intentionally left out. Only those panels most relevant to creating a file definition are shown. To define this file definition, you need to take the following steps:

1. Identify the file type.
2. Describe the structure of the file.
3. Provide field information.
4. Provide column headings for use on reports.

Figure 1-13 shows how to specify the file type in the File Options panel.

FILFD----- MARKISPF . ISPLAG1 . SOURCE(EMPFILE) -----  
OPTION ==> 3  

	FILE	OPTIONS	FILE
1	PROLOGUE	- File prologue	: EMPFILE
2	RELATIONAL	- DB2 or SQL/DS based tables	
3	KSDS	- VSAM Key Sequenced Data Set	
4	ESDS	- VSAM Entry Sequenced Data Set	
5	RRDS	- VSAM Relative Record Data Set	
6	AIX	- VSAM Alternate Index	
7	DLI	- DL/I data base	
8	DLIHDAM	- HDAM DL/I data base	
9	MSDB	- IMS Main Storage Data Base (fast path)	
10	DEDB	- IMS Data Entry Data Base (fast path)	
11	ISAMFIX	- Fixed length ISAM file	
12	ISAMVAR	- Variable length ISAM file	
13	FIXED	- Fixed length file	
14	VARIABLE	- Variable length file	
15	UNDEFINED	- Undefined (unformatted) file	
16	GDBI	- Generalized Data Base Interface	
17	PACKED	- Compressed file	

Figure 1-13 Identifying the File Type

Note that any one of 16 different file types are available on this panel. This example assumes that the file is a VSAM key sequenced data set (KSDS). Select KSDS (option 3).

The next panel that automatically appears enables you to define the structure of the file. The File Definition panel is illustrated in Figure 1-14.

**Note:** The title of a File Definition panel is prefaced by the file type, such as VSAM KSDS, Relational, Fixed, and so on.

FILNALSU----- MARKISPF . ISPLAG1 . SOURCE(EMPFILE) ----- ROW 1 OF 13  
COMMAND ==>  

VSAM KSDS FILE DEFINITION				
BUFFER SIZE ==> 80		FILE ID	==> TABLE1	
GLOSSARY ==> _		EXPIRATION DATE	==> _ / _ / _	
		USER DATA	==> _____	
Line	Segment	Hierarchical	Fixed	Segment
Cmd	Name	Level	Occurrences	Order
''''	EMPLOYEE	1	__	A
''''	_____	-	__	-
''''	_____	-	__	-
''''	_____	-	__	-
''''	_____	-	__	-
''''	_____	-	__	-
''''	_____	-	__	-
''''	_____	-	__	-
''''	_____	-	__	-
''''	_____	-	__	-

Figure 1-14 Defining the File Structure

Extremely complex file structures of up to 9 hierarchical levels and up to 255 different segment types can be defined. Relational tables and relational views can be joined in a virtually unlimited number of ways. In this example, you use a single level, flat file structure.

First, define the buffer size and provide a name for the single segment involved. This is the root level segment or level 1 segment. Subordinate segments can be identified at levels 2, 3, 4, and so on. Selecting the root segment takes you to the Field Definitions panel illustrated in [Figure 1-15](#).

```

FILNUL0F----- ISPLCB1.MISPF.SOURCE(EMPFILE) ----- ROW 1 OF 7
COMMAND ==>
FIELD DEFINITIONS
HIERARCHICAL LEVEL ==> 1      SEGMENT NUMBER ==> 001
FIXED OCCURRENCES ==> _____ SEGMENT ORDER ==> A
SCROLL ==> CSR
FILE : EMPFILE
SEGMENT: EMPLOYEE

Line  Field Name      ... Field ...  Dec  Seg
Cmd  Alternate (long) Name  Loc Len Typ Plc Rnd  Key
.... EMPNUM           1   8   C   _   _   1
.... EMPNAME          9   20  C   _   _   _
.... EMPTITLE         29   20  C   _   _   _
.... EMPSAL           49   6   Z   _   _   _
.... EMPSEX           55   1   C   _   _   _
.... EMPAGE           56   2   C   _   _   _
.... EMPFILL          58   23  C   _   _   _

```

Figure 1-15 Defining the Fields

The Field Definitions panel allows you to define fields by providing the field names, location, length, and data type of the fields involved in the file. From this panel, there are a lot more specifications that could be provided for a different variety of field types, and editing characteristics required for reports. Defaults are assumed for the editing characteristics, so you need only provide the minimum necessary data. Selection of any one of the fields automatically brings up the Field Column Headings panel illustrated in [Figure 1-16](#).

FILAINFO --- ISPLCB1.MISPF.SOURCE(EMPFILE) ----- ROW 1 OF 9  
COMMAND ==>

ADDITIONAL FIELD INFORMATION FOR FIELD: EMPNUM

Output Field Length ==> \_ Floating Character ==> \_  
Fill Character ==> \_ Trailing Character ==> \_

Alternate Field Name ==> \_ <={Long Name}  
External Field Name ==> \_ <=={(DB2 Column name or  
IMS DBD Field Name)

Field Description  
==> \_\_\_\_\_

Line	Cmd	Column Heading Text
....		EMPLOYEE
....		NUMBER
....		_____
....		_____
....		_____
....		_____
....		_____
....		_____
....		_____

Figure 1-16 Defining Column Headings

The Field Column Headings panel shown in [Figure 1-16](#) enables you to provide column headings as they are to appear on a printed report. Nine lines of column heading can be provided.

That is all there is to it. You have created a file definition.

All of the file definition types supported by VISION:Builder, VISION:Inform, and VISION:Transact are defined in a similar way. You simply select the file definition type that you would like to create from the File Options panel shown in [Figure 1-13](#).

In addition to creating file definitions, you can also use VISION:Workbench for ISPF to create logical data views, arrays, tables, transactions, and screen definitions.

## Using the VISION:Workbench for ISPF Utilities

VISION:Workbench for ISPF provides you with the ability to perform utility functions interactively. The Utility subsystem of VISION:Workbench for ISPF is easy to use. It helps you maintain your VISION:Builder and VISION:Transact libraries by performing such functions as:

- Condensing libraries
- Initializing libraries
- Backing up libraries
- Restoring libraries.

Further, it provides an easy means of manipulating cataloged items on the library:

- Delete items
- Copy items
- Document items
- Retrieve items in source form.

When Utilities (option 50) are selected from the VISION:Workbench for ISPF Main menu, the Utility Selection Menu panel in [Figure 1-17](#) appears.

```

UTILSEL ----- UTILITY SELECTION MENU -----
COMMAND ==>

      COMLIB MAINTENANCE FUNCTIONS:

      1 CONDENSE   - Condense A Comlib
      2 INITIALIZE - Create / Initialize A Comlib
      3 BACKUP     - Backup A Comlib
      4 RESTORE    - Restore A Comlib

      ITEM MAINTENANCE FUNCTIONS:

      5 DELETE     - Delete Comlib Items
      6 COPY       - Copy Items From One Comlib To Another
      7 DOCUMENT   - Document Comlib Items
      8 SOURCE     - Retrieve Comlib Items In Source Form
  
```

Figure 1-17 The Utility Selection Menu Panel

On this panel, you make the necessary selection and the appropriate follow-on panels appear for you to enter any required information.

For example, selecting Initialize (option 2) to initialize a COMLIB displays the COMLIB Initialization panel shown in [Figure 1-18](#).

```

UTILINIT ----- COMLIB INITIALIZATION -----
OPTION ==>

SPECIFY THE FOLLOWING INFORMATION:

      Library Name      ==> M4LIB

      Blocking Factor   ==>
  
```

Figure 1-18 Initializing a Library

To initialize a new COMLIB, enter the information on the panel.

Backing up a COMLIB is equally as easy. Selecting Backup (option 3) displays the COMLIB Backup panel shown in [Figure 1-19](#).

```

UTILBKUP ----- COMLIB BACKUP -----
OPTION ==>

SPECIFY THE COMLIB THAT IS TO BE BACKED UP:

      Library Name ==> M4LIB

SPECIFY DATASET TO BACKUP TO:

      Dataset Name ==> 'ISPLAG1.M4LIB.DUMP'
    
```

Figure 1-19 Backing Up a Library

On this panel, you specify what library to backup and where to put it.

Copying items from one library to another is also a very powerful capability. To do this, select Copy (option 6) from the Utility Selection Menu panel to display the follow-on Copy Utility panel shown in [Figure 1-20](#).

```

UTILCOPY----- COPY UTILITY -----
OPTION ==>

      Blank - Display Item List      G - Transaction Groups
      A - Array Definitions          Q - Request Groups
      D - Logical Data Views         R - Requests
      F - File Definitions           S - Screen Definitions
                                   T - Table Definitions

SPECIFY "FROM" COMLIB BELOW, THEN PRESS ENTER TO SPECIFY "TO" COMLIB

Comlib Name ==> 'ISPDLD1.M4LIB'

Item Name   ==>                (Leave Blank For Member List Display)

Master File Name ==>            (Specify When Copying Transaction Groups)

Generic Item   ==>                (Yes Or No)
    
```

Figure 1-20 Specifying Items to Copy

Here, you provide the identification of the item you want to copy. The codes for the items are listed on the panel. Specify where to get the item (COMLIB name) and the name of the item that you are copying. On the next panel, shown in [Figure 1-21](#), you specify where you want the item to go.

```

UTILCOP2- FROM ISPDLD1.M4LIB -----
COMMAND ==>

SPECIFY "TO" COMLIB:

Library Name      ==> 'ISPDLD1.M4LIB2'

Replace Option    ==> NO

  No   - Suppress Copy If Items Already Exists In "To" Comlib
  Yes  - Replace Like-Named Items In "To" Comlib

Replace Rule      ==>

  Blank - Replace Like-Named Items
  Old   - Replace Only If Like-Name Item Has Older Date
  New   - Replace Only If Like-Name Item Has Most Recent Date

```

Figure 1-21 Specifying Where Copied Items

On the Copy Destination or Specify To COMLIB panel shown in [Figure 1-21](#), there are many features to protect you from situations where an item of an identical name already exists. Required entries are explained on the panel itself and in the Help panels.

## Using the Help Facility

There are two Help facilities that are provided with VISION:Workbench for ISPF. Both Help facilities provide help within the context of what you are doing. The two Help facilities that are provided are:

- Panel Level      –    for help concerning the entire panel.
- Field Level     –    for help concerning a specific field.

Figure 1-22 shows a panel and the corresponding Help panel that is displayed when you enter the HELP primary command or press the Help key (usually PF1).

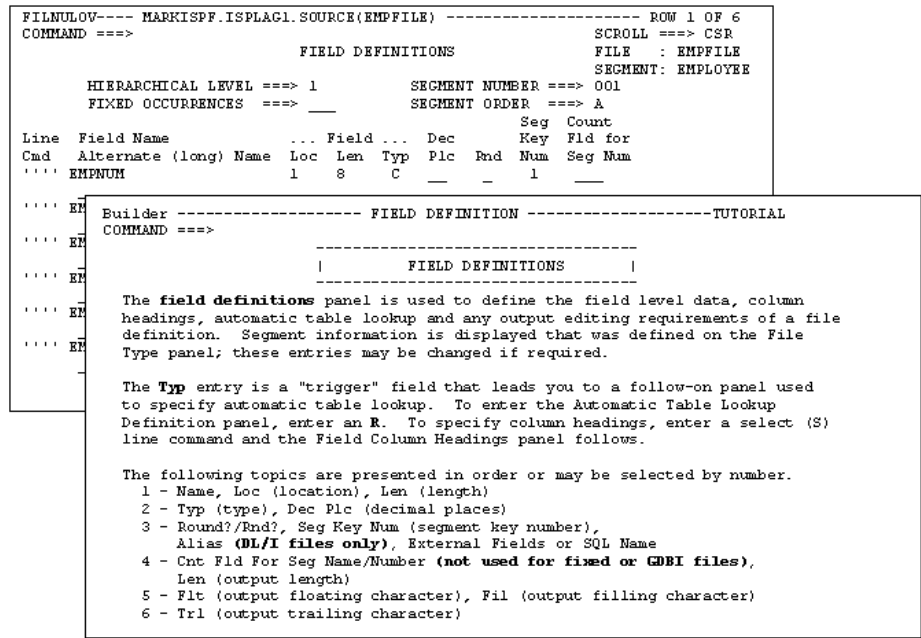


Figure 1-22 Requesting Panel Level Help



Figure 1-23 shows a request for field level help (?) and the Help panel that is displayed.

```

FILNUL OF---- ISPLCB1.MISPF.SOURCE(EMPFILE) ----- ROW 1 OF 7
COMMAND ==>
FIELD DEFINITIONS
HIERARCHICAL LEVEL ==> 1          SEGMENT NUMBER ==> 001
FIXED OCCURRENCES ==> ____      SEGMENT ORDER ==> A
FILE : EMPFILE
SEGMENT: EMPLOYEE

Line  Field Name      ... Field ...  Dec  Seg
Cmd  Alternate (long) Name  Loc Len Typ Plc Rnd  Key
.... EMPNUM           1    8    ?   _   _   1
.... EMPNAME          9   20   C   _   _   -
.... EMPTITLE         29   20   C   _   _   -

.... EMP Builder ----- FIELD DEFINITION ----- TUTORIAL
.... EMP COMMAND ==>
.... EMP The Typ entry is the type of data in the field. Enter one of the following:
.... EMP C or blank - Character string (use C for a virtual key field).
.... EMP Z          - Zoned decimal number.
.... EMP P          - Packed decimal number.
.... EMP F          - Fixed point binary number.
.... EMP D          - Date Field - Lilian Date Format.
.... EMP E          - Floating point number.
.... EMP U          - Variable length.
.... EMP L,S        - Time processing.
.... EMP R          - Automatic table lookup result field. An R entry leads you to
                      a follow-on panel that is used to define automatic table
                      lookup. The field name, field length, etc is carried forward.

The Dec Plc (decimal places) is used to specify the number of decimal places
in a numeric field or time constant. For packed fields enter 0-15. For all
other field types enter 0 - 9. Leave blank for type U fields.

```

Figure 1-23 Requesting Field Level Help

VISION:Workbench for ISPF provides more than 700 panels of help information.

You can immediately get help for the information you need. Merely use the HELP primary command to get panel level help or enter a ? in a field to get field level help. Use the END primary command to return to your application.



# Using VISION:Workbench for ISPF

---

This chapter teaches you how to use VISION:Workbench for ISPF. Reading it enables you to quickly start using VISION:Workbench for ISPF in a productive and efficient manner.

[Similarities to ISPF on page 2-1](#) briefly discusses basic information that is useful while learning VISION:Workbench for ISPF. It introduces you to some common VISION:Workbench for ISPF terminology. It also provides general background information to help you start using VISION:Workbench for ISPF.

The rest of this chapter discusses detailed information about using VISION:Workbench for ISPF and concepts behind VISION:Workbench for ISPF. Some of this information is specific in nature and may not apply to all parts of VISION:Workbench for ISPF. You can determine which sections are of interest to you by scanning the [Chapter . Contents](#).

## Similarities to ISPF

VISION:Workbench is designed to look and function in a manner very similar to ISPF. Because of this, users that are already familiar with ISPF find themselves right at home using VISION:Workbench. VISION:Workbench's panel formats, command entry, and interactive Help facility are all patterned after ISPF.

The only major difference in the way ISPF and VISION:Workbench function is in the structured panel hierarchy that VISION:Workbench creates and maintains. This hierarchy allows you to construct your applications and definitions in a structured and organized manner. When using VISION:Workbench for ISPF to edit your source, you proceed through the VISION:Workbench for ISPF panel hierarchy using the Select line command. You travel back up through the hierarchy using the END primary command.

[Figure 2-1](#) illustrates the concept of panel hierarchies as it applies to the VISION:Workbench for ISPF File Definition subsystem. Notice that the entries become more detailed and specific as you travel down the hierarchy.

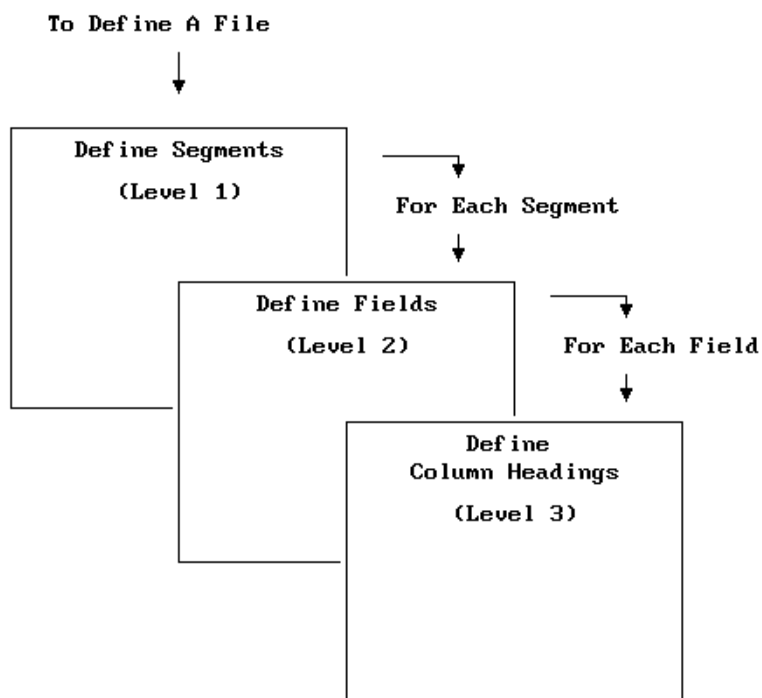


Figure 2-1 File Definition Subsystem Panel Hierarchy



## VISION:Workbench for ISPF Screen Formats

Figure 2-3 shows the typical format of a VISION:Workbench data entry panel. If you are familiar with ISPF, you probably noticed that this format is very similar to the standard ISPF panel format.

FILNALSV ---- MARKISPF . ISPLAG1 . SOURCE (EMPFILE) ----- ROW 1 OF 8  
COMMAND ==> ----- SCROLL ==> CSR

VSAM KSDS FILE DEFINITION FILE: EMPFILE

BUFFER SIZE ==> \_\_\_\_\_ FILE ID ==> \_\_\_\_\_  
GLOSSARY ==> - EXPIRATION DATE ==> \_/\_/\_  
USER DATA ==> \_\_\_\_\_

LINE CMD	SEGMENT NAME	HIERARCHICAL LEVEL	FIXED OCCURRENCES	SEGMENT ORDER
''''	_____	---	_____	---
''''	_____	---	_____	---
''''	_____	---	_____	---
''''	_____	---	_____	---
''''	_____	---	_____	---
''''	_____	---	_____	---
''''	_____	---	_____	---
''''	_____	---	_____	---

\*\*\*\*\* BOTTOM OF DATA \*\*\*\*\*

Figure 2-3 Typical Panel Format

The first section of a VISION:Workbench for ISPF panel is called the header section. The header section consists of the following three lines:

- Title line.
- Command line.
- Message line.

Among other things, the title line provides panel identification information.

The command line is where you enter menu selections and/or VISION:Workbench for ISPF primary commands.

The last line in the header section is the message line. VISION:Workbench for ISPF uses this line to display informational messages.

All VISION:Workbench for ISPF panels contain a header section.

The remaining two sections, shown in Figure 2-3, comprise the data entry areas of a panel. VISION:Workbench for ISPF utilizes the following types of data entry areas:

- Fixed data entry areas.
- Scrollable data entry areas.

The fixed data entry area contains entries that occur only once. These fields usually contain summary information from the previous panel (also known as the parent panel).

In contrast to the fixed data entry area, the scrollable data entry area contains multiply-occurring data rows that pertain to the current panel. When entering information in scrollable data areas, the ISPF UP and DOWN scroll commands can be used to page through the multiply-occurring rows of data. The ISPF LEFT and RIGHT scroll commands can be used to display scrollable areas that are wider than your terminal.

## Panel Name

For ease of reference, all VISION:Workbench for ISPF data entry panels have been assigned a unique panel name. This name is displayed in the left corner of the panel title line. [Figure 2-4](#) has the panel name PRIMOPT4.

```
PRIMOPT4 ----- VISION:Builder DEVELOPMENT FACILITY -----
OPTION ==>

SPECIFY/REVIEW VISION:Workbench PARAMETERS:

  10 PARAMETERS - Specify Or Display Parameters

EDIT VISION:Builder APPLICATIONS:          EDIT COMLIB DEFINITIONS / UTILITIES:

  20 APPLICATION - Edit Application          40 FILE           - Edit File Def
  21 PROCEDURE   - Edit Procedure           41 LDU            - Edit LDU Def
  25 REQUEST     - Edit Request             42 ARRAY          - Edit Array Def
                                           43 TABLE         - Edit Table Def
  30 FOREGROUND  - Execute Foreground       44 TRANSACTION   - Edit Tran Def
  31 BACKGROUND  - Submit Background        50 UTILITIES     - Access COMLIB
                                           51 IMPORT        - Quick Starts

VISION:Workbench FOR ISPF DEVELOPMENT FACILITY:
  T TUTORIAL    - View The VISION:Workbench Tutorial
  X EXIT        - Exit VISION:Workbench
```

Figure 2-4 Displaying the Panel Name

The panel titled “Development Facility” is referred to as the Main menu or the Primary Option menu in this guide.

Panel names are generally eight characters long. They consist of a prefix, usually four characters long, followed by a suffix, which makes up the remainder of the name. The following table lists all of the panel name prefixes that are used by VISION:Workbench for ISPF. Notice how each of these prefixes identifies the particular subsystem to which the panel belongs.

Panel Name Prefix	Subsystem Description
PRIM	Product Development/Administration Facility (Main menu or Primary Option menu)
PARM	Parameters subsystem
UTIL	Utilities subsystem
ITM	Item Entry subsystem
VAL	Validation subsystem
SAVE	Save subsystem
FIL	File Definition subsystem
LDV	Logical Data View Definition subsystem
ARR	Array Definition subsystem
TBL	Definition subsystem
TRN	Transaction Definition subsystem
APP4	VISION:Builder Application subsystem
ASL4	VISION:Builder ASL Free-Form statements
REQ4	VISION:Builder Request subsystem
BAT	VISION:Builder Batch subsystem
FOR	VISION:Builder Foreground Execution subsystem
APP5	VISION:Transact Application Subsystem
ASL5	VISION:Transact ASL Free-Form Statements



Panel Name Prefix	Subsystem Description
REQ5	VISION:Transact Request Subsystem
SCRN	VISION:Transact Screen Developer Subsystem
GEN	VISION:Transact Generate Subsystem

The remaining portion of the panel name, called the suffix, uniquely identifies a specific panel within the subsystem. The suffix usually represents the specific function that the panel performs within the subsystem.

In the following text, panels can be referenced by the fixed portion of the title. For example, “an Item Entry panel” may reference a Definition Item Entry panel, an Application Item Entry panel, or a File Item Entry panel. Panel titles are in mixed case and do not include dashes.

Another means of uniquely identifying a panel is through the panel identification or panel ID. The panel ID should not be confused with the panel name. While the panel name provides an EXTERNAL means for the USER to uniquely identify each panel, the panel ID provides an INTERNAL means for ISPF or VISION:Workbench for ISPF to uniquely identify each panel.

To view the panel ID, rather than the panel name, in the upper-left corner of a panel, enter the PANELID command on the primary command line (second line) of any panel.

```

M9EYAPBD ----- VISION:Builder DEVELOPMENT FACILITY -----
OPTION ==> PANELID

SPECIFY/REVIEW VISION:Workbench PARAMETERS:

  10 PARAMETERS - Specify Or Display Parameters

EDIT VISION:Builder APPLICATIONS:          EDIT COMLIB DEFINITIONS / UTILITIES:

  20 APPLICATION - Edit Application          40 FILE          - Edit File Def
  21 PROCEDURE  - Edit Procedure            41 LDU           - Edit LDU Def
  25 REQUEST    - Edit Request              42 ARRAY         - Edit Array Def
                                           43 TABLE        - Edit Table Def
  30 FOREGROUND - Execute Foreground        44 TRANSACTION  - Edit Tran Def
  31 BACKGROUND - Submit Background          50 UTILITIES    - Access COMLIB
                                           51 IMPORT       - Quick Starts

VISION:Workbench FOR ISPF DEVELOPMENT FACILITY:
  T TUTORIAL   - View The VISION:Workbench Tutorial
  X EXIT       - Exit VISION:Workbench

```

Figure 2-5 Using the PANELID Command

To turn the panel ID display off and return to the panel name display, reenter the PANELID command on any panel primary command line.

# Commands

As with ISPF, VISION:Workbench for ISPF commands can be broken down into the following two categories:

- Primary commands.
- Line commands.

The samples below show where these different types of commands are entered on a VISION:Workbench for ISPF panel.

- Primary commands are entered on the second line, after COMMAND==>.
- Line commands are entered in the lines beneath Line Cmd. The line command field is always the left-most field in a scrollable data row.

FILNULSF----- ISPLCB1.MISPF.SOURCE(EMPLOYEE) ----- ROW 1 OF 10  
COMMAND ==>  SCROLL ==> DATA  
FIXED LENGTH FILE DEFINITION FILE : EMPLOYEE  
  
RECORD LENGTH ==> \_\_\_\_\_ FILE ID ==> \_\_\_\_\_  
RECORDS PER BLOCK ==> \_\_\_\_\_ EXPIRATION DATE ==> \_\_ / \_\_ / \_\_  
BUFFER SIZE ==> 1 USER DATA ==> \_\_\_\_\_  
GLOSSARY ==> -  
  

Line Cmd	Segment Name	Hierarchical Level	Segment Number	Fixed Occurrences	Segment Order
''''	_____	-	_____	_____	-
''''	_____	-	_____	_____	-
''''	_____	-	_____	_____	-
''''	_____	-	_____	_____	-
''''	_____	-	_____	_____	-
''''	_____	-	_____	_____	-
''''	_____	-	_____	_____	-
''''	_____	-	_____	_____	-
''''	_____	-	_____	_____	-
''''	_____	-	_____	_____	-

  
\*\*\*\*\* BOTTOM OF DATA \*\*\*\*\*

Figure 2-6      Entering Commands in VISION:Workbench

## Primary Commands

Primary command entry in VISION:Workbench works the same as primary command entry in ISPF. In fact, most ISPF primary commands are also active while in VISION:Workbench. The following table lists some of the more common ISPF primary commands and summarizes how each of these commands work within VISION:Workbench.

ISPF Command Name	Default PF Key	ISPF Primary Command Description as it Pertains to VISION:Workbench for ISPF
=option	N/A	Allows you to jump directly from one menu option to another.
CANCEL	N/A	Ends the current edit session WITHOUT SAVING the item. Any changes that have been made to the item are lost.
DOWN	PF8/20	Used within a scrollable data section to page toward the bottom of the data.
END	PF3/15	Ends the current operation. In most cases, using the END command returns you to the previous panel.
HELP	PF1/13	Displays a Help panel that pertains to the panel you are currently working on.
KEYS	N/A	Displays a panel that lets you change your PF key definitions.
LEFT	PF10/22	Used within a scrollable data section to page toward the left margin.
LOCATE	N/A	Used to locate a specific member on a Member or Item Selection List display.
PANELID	N/A	Used to turn the Panel ID display on and off.
PRINT	N/A	Copies an image of your display screen to your ISPF list data set.
RESET	N/A	Resets all Line command fields back to the Line command pad character that is specified in your VISION:Workbench for ISPF session parameters.
RETURN	PF4/16	Returns you to the last Primary Option menu that was displayed.
RIGHT	PF11/23	Used within a scrollable data section to page toward the right margin.

ISPF Command Name	Default PF Key	ISPF Primary Command Description as it Pertains to VISION:Workbench for ISPF
SAVE	N/A	Allows you to validate and save the item you are editing without ending your edit session. A parameter of "ASIS" can be used with this command to bypass validation and go directly to "Save" processing.
SPLIT	PF2/14	Splits the display screen into two logical screens. Due to the possibility of ddname conflict between multiple ISPF sessions, this command is not active while in the VISION:Workbench for ISPF Utilities subsystem or when editing an item directly from a COMLIB.
SWAP	PF9/12	Used during split screen mode to position the cursor in another logical screen.
UP	PF7/19	Used within a scrollable data section to page toward the top of the data.

As with ISPF, VISION:Workbench primary commands can be entered in one of the following ways:

- By entering the command on the command line and pressing the Enter key.
- By pressing a PF key that has been defined to execute a specific command.

You can use the KEYS primary command to view and/or change your current PF key (program function key) assignments.

Whenever you start a VISION:Workbench for ISPF session, your PF key assignments are identical to the PF key assignments of your last VISION:Workbench for ISPF session. VISION:Workbench for ISPF "remembers" any changes you make to your PF key assignments.

Changing your PF key assignments during your VISION:Workbench for ISPF session does not affect your original ISPF PF key assignments.

The following table lists and summarizes additional primary commands that are specific to VISION:Workbench for ISPF. Notice that some of these commands have restrictions in that they are only active in certain areas of VISION:Workbench for ISPF or under certain conditions during your VISION:Workbench for ISPF session. These commands do not have default PF key assignments.

Primary Command	VISION:Workbench for ISPF Primary Command Description
DISPATTR	Allows you to temporarily jump from the current option to the Screen Developer DISPATTR option. The END command should be used to return to the original option. This command is only active in the Screen Developer subsystem.
DUP ON   OFF	DUP ON activates duplicate name checking, while DUP OFF disables duplicate name checking. Each VISION:Workbench for ISPF session begins with duplicate name checking ON.
FIELDS	Allows you to temporarily jump from the current option to the Screen Developer FIELDS option. The END command should be used to return to the original option. This command is only active in the Screen Developer subsystem.
NAMES F	Allows you to view a series of panels which list all valid VISION:Builder flag names, along with a description of each flag. This command is specific to the VISION:Builder portion of VISION:Workbench for ISPF.
NAMES HELP	This command (or NAMES ?) displays the tutorial panels describing the various forms of the NAMES command. This command is specific to the VISION:Builder edit portion of VISION:Workbench for ISPF.
NAMES ON   OFF	NAMES ON activates definition checking, while NAMES OFF disables definition checking. The effect of the command lasts until it is reversed or until the end of the your VISION:Builder application edit session.
NAMES q	Displays a list of valid field names for any file or array in your VISION:Builder application (q is the qualifier of the file or array field list to be displayed). Once the list is displayed, you can select one of the field names. Selecting a field name returns you to the original screen with the selected field displayed on the message line.

Primary Command	VISION:Workbench for ISPF Primary Command Description
NAMES STATUS	Displays information pertinent to the VISION:Builder definition checking functions, including a list of run files which are subject to definition checking. For each file, the display shows the RF usage, the RF name, the location of the definition (instream, COMLIB, not found) and whether this is a file or an array. The current status of definition checking (ON or OFF) is also shown.
NOCHECK	This command is specific to the Procedure subsystem. It allows you to bypass validation processing when exiting from this subsystem.
PROCATTR	Allows you to temporarily jump from the current option to the Screen Developer PROCATTR option. The END command should be used to return to the original option. This command is only active in the Screen Developer subsystem.
PROFILE	Allows you to temporarily jump from the current option to the Screen Developer PROFILE option. The END command should be used to return to the original option. This command is only active in the Screen Developer subsystem.
PROLOGUE	Allows you to temporarily jump from the current option to the Screen Developer PROLOGUE option. The END command should be used to return to the original option. This command is only active in the Screen Developer subsystem.
SEGMENTS	Allows you to temporarily jump from the current option to the Screen Developer SEGMENTS option. The END command should be used to return to the original option. This command is only active in the Screen Developer subsystem.

Primary Command	VISION:Workbench for ISPF Primary Command Description
SELECT	This command should not be confused with the Select line command. The SELECT primary command is used to invoke “Zoom” processing for a fixed data entry field. This command is only active in specific options of the Screen Developer subsystem.
SHOWFLDS ON SHOWFLDS OFF	The SHOWFLDS command is used to turn the display of screen layouts data fields on and off. This command is only active in the Screen Developer subsystem.
VIEW	Allows you to temporarily jump from the current option to the Screen Developer VIEW option. The END command should be used to return to the original option. This command is only active in the Screen Developer subsystem.

## Line Commands

Line commands differ from primary commands in that they serve a very specific purpose. They are used to manipulate and/or process the scrollable data entry area of a panel. Line commands are entered in the line command field of scrollable data rows. The line command field is always the left-most field in a scrollable data row.

The following table lists the line commands that are supported by VISION:Workbench for ISPF.

Line Command	VISION:Workbench for ISPF Line Command Description
S	Selects a scrollable data row for further processing by invoking the next lower level panel.
I	Insert a blank line after this line.
In	Insert “n” blank lines after this line.
D	Delete this line.
Dn	Delete “n” lines.
DD	Delimits a block of lines to be deleted.
R	Repeat this line.
Rn	Repeat “n” lines.
RR	Delimits a block of lines to be repeated.
C	Copy this line.

Line Command	VISION:Workbench for ISPF Line Command Description
Cn	Copy “n” lines.
CC	Delimits a block of lines to be copied.
M	Move this line.
Mn	Move “n” lines.
MM	Delimits a block of lines to be moved.
A	Identifies the line after which the moved/copied lines are to be inserted.
An	Inserts “n” copies of the moved/copied lines after this line.
B	Identifies the line before which the moved/copied lines are to be inserted.
Bn	Inserts “n” copies of the moved/copied lines before this line.
P	Passes this line during definition checking. Field names on this line are accepted as valid for the remainder of the application edit session. Do not perform name validation on this field.

VISION:Workbench line commands differ from ISPF line commands. Due to the panel hierarchy, VISION:Workbench line commands process the entire data hierarchy below the row being processed (rather than just processing the current scrollable row). The following example uses the VISION:Workbench File Definition subsystem to illustrate this concept.

Assume you are editing the Employee file definition. From the File Definition panel, enter S, for the Select line command, to choose the Employee segment as shown below.

```

FILNULSF---- ISPLCB1.MISPF.SOURCE(EMPFILE) ----- ROW 1 OF 1
COMMAND ==>                                     SCROLL ==> DATA
                                     FIXED LENGTH FILE DEFINITION  FILE : EMPFILE

      RECORD LENGTH    ==> 3020          FILE ID          ==> _____
      RECORDS PER BLOCK ==> _____    EXPIRATION DATE ==> __ / __ / __
      BUFFER SIZE      ==> _____    USER DATA      ==> _____
      GLOSSARY         ==> _

Line   Segment   Hierarchical   Segment   Fixed   Segment
Cmd    Name      Level          Number    Occurrences  Order
S'''   EMPLOYEE   1              01        _____  A
***** BOTTOM OF DATA *****

```

Figure 2-7 Selecting a Data Row for Further Processing



This enables you to view the fields that have been defined for this segment.

FILNUL0F---- ISPLCB1.MISPF.SOURCE(EMPFILE) -----										ROW 1 OF 12
COMMAND ==>										SCROLL ==> DATA
FIELD DEFINITIONS										FILE : EMPFILE
										SEGMENT: EMPLOYEE
HIERARCHICAL LEVEL ==> _					SEGMENT NUMBER ==> 01					
FIXED OCCURRENCES ==> _					SEGMENT ORDER ==> A					
Line							Seg			
Cmd	Name	Loc	Len	Typ	Plc	Rnd	Key Num	External Field name	Output Len	Edit Flt Fil Tr
'''	EMPNUM	1	8	C			1			
'''	EMPNAME	9	20	C						
'''	EMPTITLE	29	20	C						
'''	EMPSAL	49	6	Z						
'''	EMPSEX	55	1	C						
'''	EMPDIV	56	2	C						
'''	EMPFILL	58	23	C						
'''										
'''										
'''										
'''										
'''										
***** BOTTOM OF DATA *****										

Figure 2-8 Viewing the Fields Defined for a Segment

Returning to the File Definition panel, enter R, for Repeat line command, for the Employee segment just selected.

```

FILNULSF---- ISPLCB1.MISPF.SOURCE(EMPFILE) ----- ROW 1 OF 1
COMMAND ==> SCROLL ==> DATA
                FIXED LENGTH FILE DEFINITION      FILE : EMPFILE

RECORD LENGTH   ==> 3020          FILE ID           ==> _____
RECORDS PER BLOCK ==> _____ EXPIRATION DATE ==> __ / __ / __
BUFFER SIZE     ==> _____  USER DATA        ==> _____
GLOSSARY        ==> -

Line   Segment   Hierarchical   Segment   Fixed   Segment
Cmd    Name      Level         Number    Occurrences   Order
R'''   EMPLOYEE   1             01        _____   A
***** BOTTOM OF DATA *****

```

Figure 2-9 Using the Repeat Line Command

The result of the Repeat line command is shown below. Notice that you now have two identical Employee segments showing on the File Definition panel.

FILNULSF---- ISPLCB1.MISPF.SOURCE(EMPFILE) -----					ROW 1 OF 2
COMMAND ==>					SCROLL ==> DATA
FIXED LENGTH FILE DEFINITION					FILE : EMPFILE
RECORD LENGTH	==>	3020	FILE ID	==>	_____
RECORDS PER BLOCK	==>	_____	EXPIRATION DATE	==>	__ / __ / __
BUFFER SIZE	==>	_____	USER DATA	==>	_____
GLOSSARY	==>	_			
Line	Segment	Hierarchical	Segment	Fixed	Segment
Cmd	Name	Level	Number	Occurrences	Order
''''	EMPLOYEE	1	01	_____	A
''''	EMPLOYEE	1	01	_____	A
***** BOTTOM OF DATA *****					

Figure 2-10 Results of a Repeated Line

If you use the Select line command on the new, repeated version of the Employee segment, you will see that all of the field definitions belonging to the original Employee segment are repeated in the new Employee segment. (You must enter the DUP OFF primary command first or you will receive an error message.)

Of course, the “R” action discussed above has resulted in an illegal file definition because segment and field names must be unique within a definition. This is only an illustration of a concept; presumably further editing would be performed to make the definition valid.

## Interactive Help Facilities

VISION:Workbench for ISPF provides you with a complete interactive Tutorial and Help facility. This facility works in the same manner as the ISPF Help and Tutorial facility.

The Tutorial and Help facilities are really one and the same in that they contain the same information. The difference between them is how you use them. The tutorial, which you can enter from the Primary Option menus, provides you with an introduction and overview of VISION:Workbench for ISPF. If you sequentially read through the entire VISION:Workbench for ISPF tutorial, following every leg and path within the tutorial, you see every Help panel as well. The key to the difference between the Tutorial and Help facilities is the access:

- The HELP command allows you to immediately invoke a Help panel that specifically pertains to the panel you are viewing. (The ISPF default PF key for the HELP command is PF1.)
- The tutorial starts at the beginning of the Help facility and proceeds through all of VISION:Workbench's functionality.

Using the END command from a Help panel returns you to the panel from which you invoked the Help facility.

To view the Help panel for the panel you are currently using, enter the HELP as a primary command:

```

FILNUL0F----- ISPLCB1.MISPF.SOURCE(EMPFILE) ----- ROW 1 OF 7
COMMAND ==> HELP                                     SCROLL ==> CSR
                                                    FILE  : EMPFILE
                                                    SEGMENT: EMPLOYEE

                        FIELD DEFINITIONS
                        HIERARCHICAL LEVEL ==> 1          SEGMENT NUMBER ==> 001
                        FIXED OCCURRENCES ==> ____        SEGMENT ORDER  ==> A

Line  Field Name      ... Field ...  Dec      Seg
Cmd  Alternate (long) Name  Loc Len Typ Plc Rnd  Key
.... EMPNUM           1    8   C    _    _    1
.... EMPNAME          9    20   C    _    _    _
.... EMPTITLE         29    20   C    _    _    _
.... EMPSAL           49    6    Z    _    _    _
.... EMPSEX           55    1    C    _    _    _
.... EMPAGE           56    2    C    _    _    _
.... EMPFILL          58    23   C    _    _    _

```

Figure 2-11 Using the HELP Command

From the Field Definition panel, the following Help panel is displayed.

```

Builder ----- FIELD DEFINITION -----TUTORIAL
COMMAND ==>

                        | FIELD DEFINITIONS |
                        |-----|
The field definitions panel is used to define the field level data, column
headings, automatic table lookup and any output editing requirements of a file
definition. Segment information is displayed that was defined on the File
Type panel; these entries may be changed if required.

The Typ entry is a "trigger" field that leads you to a follow-on panel used
to specify automatic table lookup. To enter the Automatic Table Lookup
Definition panel, enter an R. To specify column headings, enter a select (S)
line command and the Field Column Headings panel follows.

The following topics are presented in order or may be selected by number.
1 - Name, Loc (location), Len (length)
2 - Typ (type), Dec Plc (decimal places)
3 - Round?/Rnd?, Seg Key Num (segment key number),
  Alias (DL/I files only), External Fields or SQL Name
4 - Cnt Fld For Seg Name/Number (not used for fixed or COBI files),
  Len (output length)
5 - Flt (output floating character), Fil (output filling character)
6 - Trl (output trailing character)

```

Figure 2-12 Displaying a Help Panel

To give you an even finer level of Help, VISION:Workbench for ISPF also provides a field level Help facility. This Help facility is activated by entering a question mark (?) in the first position of a field. Field level Help is currently available in all the VISION:Workbench for ISPF Edit subsystems and in parts of the VISION:Transact Generate subsystem.

The following demonstrates how the field level Help facility works using the TYP entry on the Field Definitions panel as a sample. [Figure 2-13](#) shows the resulting Help panel.

FILNUL0F---- ISPLCB1.MISPF.SOURCE(EMPFILE) ----- ROW 1 OF 7  
COMMAND ==>

FIELD DEFINITIONS  
HIERARCHICAL LEVEL ==> 1  
FIXED OCCURRENCES ==> \_\_\_\_

SEGMENT NUMBER ==> 001  
SEGMENT ORDER ==> A

SCROLL ==> CSR  
FILE : EMPFILE  
SEGMENT: EMPLOYEE

Line	Field Name	...	Field	...	Dec	Seg
Cmd	Alternate (long) Name	Loc	Len	Typ	Plc	Key Num
....	EMPNUM	1	8	?	—	1
....	EMPNAME	9	20	C	—	—
....	EMPTITLE	29	20	C	—	—
....	EMPSAL	49	6	Z	—	—
....	EMPSEX	55	1	C	—	—
....	EMPAGE	56	2	C	—	—
....	EMPFILL	58	23	C	—	—

Figure 2-13 The Field Level Help Facility

Builder ----- FIELD DEFINITION ----- TUTORIAL  
COMMAND ==>

The Typ entry is the type of data in the field. Enter one of the following:

C or blank

- Character string (use C for a virtual key field).

Z

- Zoned decimal number.

P

- Packed decimal number.

F

- Fixed point binary number.

D

- Date Field - Lilian Date Format.

E

- Floating point number.

U

- Variable length.

L,S

- Time processing.

R

- Automatic table lookup result field. An R entry leads you to a follow-on panel that is used to define automatic table lookup. The field name, field length, etc is carried forward.

The Dec Plc (decimal places) is used to specify the number of decimal places in a numeric field or time constant. For packed fields enter 0-15. For all other field types enter 0 - 9. Leave blank for type U fields.

Figure 2-14 Displaying a Help Panel for a Specific Field

When you return to the Field Definitions panel, a summary message is displayed to remind you of the valid entries for the field for which you requested help. The summary message for this example is shown below.

FILNUL0F---- ISPLCB1.MISPF.SOURCE(EMPFILE) -----									
COMMAND ==>									
ENTER: BLANK/C, Z,P,F,D,E,U,R,L OR S									
HIERARCHICAL LEVEL ==> 1									
FIXED OCCURRENCES ==> ____									
SEGMENT NUMBER ==> 001									
SEGMENT ORDER ==> A									
SEGMENT: EMPLOYEE									
Line	Field Name	...	Field	...	Dec			Seg	
Cmd	Alternate (long) Name	Loc	Len	Typ	Plc	Rnd		Key	
....	EMPNUM	1	8	?	—	—		Num	1
....	EMPNAME	9	20	C	—	—			
....	EMPTITLE	29	20	C	—	—			
....	EMPSAL	49	6	Z	—	—			
....	EMPSEX	55	1	C	—	—			
....	EMPAGE	56	2	C	—	—			
....	EMPFILL	58	23	C	—	—			

Figure 2-15 Field Help Summary Message

A special command language for use within all the Help facilities is available. The following table lists these Help facility commands.

Command Name	Default PF Key	Help Facility Command Description
BACK	N/A	Return to the previous Help panel.
SKIP	N/A	Skip to next topic.
UP	N/A	Display the next level of topics.
TOP	N/A	Display the Tutorial Table of Contents.

# Split Screen Processing

Split screen processing is an ISPF facility. Using the ISPF primary command SPLIT, you can “split” your display, thus creating two separate ISPF sessions.

Split screen processing is supported by VISION:Workbench for ISPF. However, there are two conditions under which split screen processing is disabled within VISION:Workbench for ISPF:

- While in the Utilities subsystem.
- When editing an item directly from a COMLIB.

Under these conditions, VISION:Workbench for ISPF automatically disables split screen processing. This is done to avoid possible ddname conflicts that could arise between multiple sessions.

Figure 2-16 shows split screen processing in VISION:Workbench for ISPF.

```
SCRNPROF --- MCONFIG.WBISPF50.M4INPUT(MENU) -----
COMMAND ==>                                     SCROLL ==> CSR

Field Designator      ==> X          (Default Field Designator - _ )
Highlight Designator   ==> 0          (Default Highlight Designator - @ )
Device Identifier      ==>           (Default Device ID: D3270 )

Line      Attribute    Bright/
Cmd       Character    Dark      Color      Extended      Program
0001 B H B _ _
0002 b _ B _ _
0003 D D _ _ _

PRMVDEVT ----- VISION:Transact TERMINAL DEVICE TABLE ----- Row 1 to 2 of 12
COMMAND ==>                                     SCROLL ==> CSR

Default Device ID:  D3270

<===== PARAMETERS FOR IMS/DC MFS GENERATION =====>
Device      Line  Col  MFS Device Type      PF  Ext  Ext  Ext  MOD  MOD  MID  MID
ID          Line  Col  MFS Device Type      Ky  Col  Hil  Psym  PAG  IGN  IGN  NMOD
-----
D3270      24    80  {3270,2}          24  N   N   N   Y   Y   Y   N
DP24X80    24    80  {3270,2}          24  N   N   N   Y   Y   Y   N
```

Figure 2-16 Split Screen Processing

## Editing Items

Now that you have some general background information on how VISION:Workbench for ISPF works, you can use VISION:Workbench for ISPF to create and/or maintain your applications and definitions.

The various types of items supported by VISION:Workbench for ISPF are divided into two distinct categories:

- Definition items.
- Application items.

This grouping is summarized in the following table.

VISION:Workbench for ISPF		
Product	Definition Items	Application Items
VISION:Builder Support	File Definitions Logical Data Views Array Definitions Table Definitions Transaction Definitions	Applications Procedures Requests
VISION:Inform Support	File Definitions Logical Data Views Table Definitions	Procedures Requests
VISION:Transact Support	File Definitions Screen Definitions	Applications Procedures Requests

Notice that requests, procedures, and screen definitions are not considered definition items. Rather, they are categorized as application items. This distinction becomes important later in the edit process.

Now you are ready to start the edit process.

The VISION:Workbench for ISPF edit process consists of the following six steps:

1. Select an item type.
2. Specify item location and name.
3. Specify validation libraries.
4. Edit the item.
5. Validate the item.
6. Save the item.

Each of these steps is discussed in the following sections.

Step 1.: Select an Item Type

The first step in the VISION:Workbench for ISPF edit process is to select the type of item that you want to create and/or modify. This is done by entering an option number on one of the product menus shown in [Figure 2-17](#).

- PRIMOPT4 is the Main menu or Primary Option menu for VISION:Builder.
- PRIMMENU is the Main menu or Primary Option menu for VISION:Inform.
- PRIMOPT5 is the Main menu or Primary Option menu for VISION:Transact.

These menus are normally referred to as Main menus.

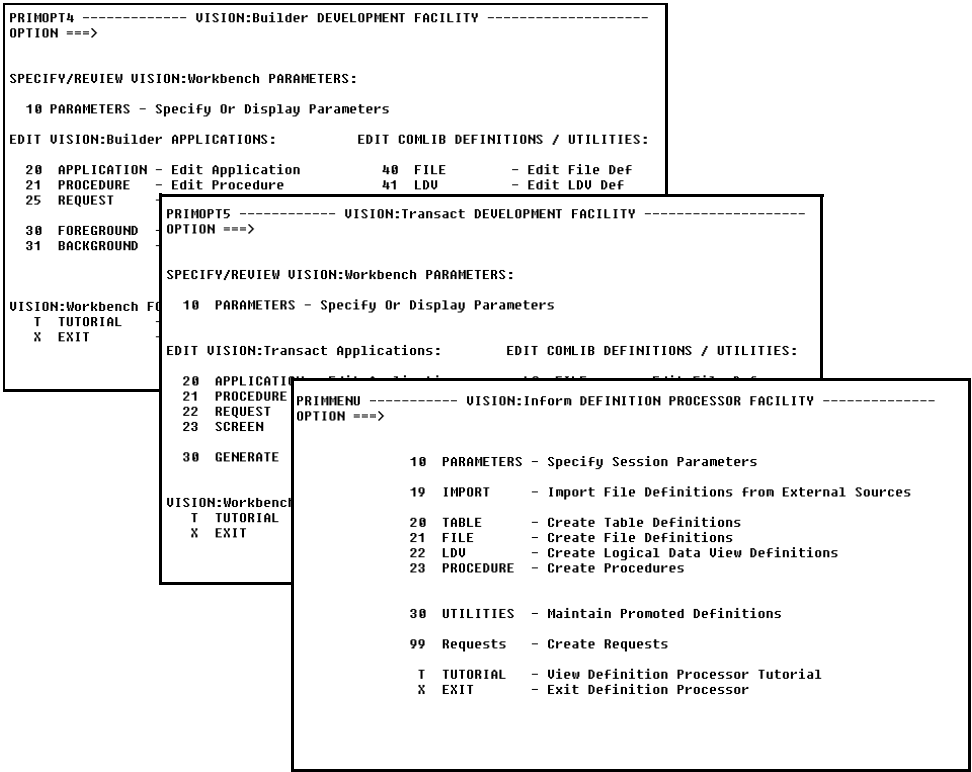


Figure 2-17 Main menus

Step 2.: Specify Item Location

Once you have selected an item type from a main menu, you automatically proceed to Step 2. of the edit process. In Step 2., you specify the source libraries and item name.

- Specifying source libraries allows you to tell VISION:Workbench for ISPF where an item is located.
- Specifying an item name allows you to tell VISION:Workbench for ISPF the name of the particular item that you want to edit.



## Specifying Source Libraries

There are two different Item Entry panels that can be used to specify source libraries, the Definition Item Entry panel and the Application Item Entry panel. These panels are shown in [Figure 2-18](#).

```

ITHENTRY ----- DEFINITION - ITEM ENTRY PANEL -----
COMMAND  ===>                                     SCROLL ===> CSR

ISPF LIBRARY:
PROJECT ===>                                     ===>
GROUP   ===>                                     ===>
TYPE    ===>                                     ===>
MEMBER  ===>                                     (Blank for member selection list)

OTHER PARTITIONED OR SEQUENTIAL DATASET:
DATASET NAME ===>
VOLUME SERIAL ===>                                     (If not cataloged)

COMLIBS:
COMLIB1  ===>
COMLIB2  ===>
COMLIB3  ===>
COMLIB4  ===>
ITEM NAME ===>                                     (Blank for member selection list)

DATASET PASSWORD                                     (If password protected)

EXCLUSIVE USE OF PRIMARY COMLIB ===> N      (Y-Yes, N- No)

```

```

ITHENTAP ----- APPLICATION - ITEM ENTRY PANEL -----
COMMAND  ===>                                     SCROLL ===> CSR

ISPF LIBRARY:
PROJECT ===>                                     ===>
GROUP   ===>                                     ===>
TYPE    ===>                                     ===>
MEMBER  ===>                                     (Blank for member selection list)

OTHER PARTITIONED OR SEQUENTIAL DATASET:
DATASET NAME ===>
VOLUME SERIAL ===>                                     (If not cataloged)

DATASET PASSWORD                                     (If password protected)

```

Figure 2-18 Specifying Source Libraries

In general, VISION:Workbench Item Entry panels work in the same manner as their ISPF counterparts.

Note that the VISION:Workbench for ISPF Item Entry panels provide up to three different areas for specifying source libraries. You can specify:

- An ISPF library.
- A standard OS/390® (MVS) partitioned or sequential data set.
- A VISION:Builder or VISION:Transact COMLIB.

When completing an Item Entry panel, remember that only one of these areas can be active at a time. If you specify libraries in more than one area, as shown in [Figure 2-19](#), the library specification closest to the bottom of the Item Entry panel is used.

```

ITEMENTRY ----- DEFINITION - ITEM ENTRY PANEL -----
COMMAND ==>                                           SCROLL ==> CSR

ISPF LIBRARY:
  PROJECT ==> MARKISPF
  GROUP   ==> ISPLAG1   ==>           ==>           ==>
  TYPE    ==> M4INPUT
  MEMBER  ==>           (Blank for member selection list)

OTHER PARTITIONED OR SEQUENTIAL DATASET:
  DATASET NAME ==> 'MARKISPF.ISPRTR1.SOURCE'
  VOLUME SERIAL ==>           (If not cataloged)

COMLIBS:
  COMLIB      ==> 'ISPLAG1.M4LIB'
  COMLIB1     ==>
  COMLIB2     ==>
  COMLIB3     ==>
  ITEM NAME   ==>           (Blank for member selection list)

DATASET PASSWORD ==>           (If password protected)
EXCLUSIVE USE OF PRIMARY COMLIB ==> N (Y-Yes, N-No)

```

Figure 2-19 A Completed Item Entry Panel

### Using the Definition Item Entry Panel

As mentioned, VISION:Workbench for ISPF uses two different Item Entry panels. The first type of Item Entry panel, the Definition Item Entry panel, is used to identify the library.

- In the ISPF LIBRARY section of the Definition Item Entry panel, you simply type in the project, group, type, and member name.
- In the “Other Partitioned or Sequential Dataset” and “COMLIBS” section, you must enclose the data set name in quotation marks or your user ID will be used as a prefix.

Definition items can be retrieved for edit directly from a COMLIB. When using COMLIBs, you can specify whether or not you want to place a data set level enqueue, or lock, on the primary COMLIB. The primary COMLIB is the first COMLIB specified. If you enqueue the primary COMLIB, it is unavailable to other TSO sessions and batch jobs. As a result, do not specify “Exclusive Use of Primary COMLIB” if a production COMLIB is being used as the primary COMLIB on the Definition Item Entry panel.

## Using the Application Item Entry Panel

The second type of Item Entry panel, the Application Item Entry panel, is used to edit application items.

Application items (which include screen definitions, requests and procedures) can only be retrieved from a partitioned or sequential data set. They cannot be retrieved directly from a COMLIB. To edit a cataloged application item such as a screen definition or request, use the VISION:Workbench for ISPF Utilities subsystem source statement retrieval option to retrieve the item and store it in a partitioned or sequential data set. You can then edit the retrieved source from that data set. Please see [Using the Utilities Subsystem on page 2-68](#), Using the Utilities Subsystem, for additional information.

## Concatenating Libraries

Both VISION:Workbench for ISPF Item Entry panels are similar in that they allow you to specify library concatenations in the File Item Entry panel as shown in [Figure 2-20](#).

```

ITEMENTRY ----- FILE - ITEM ENTRY PANEL -----
COMMAND ==>                                     SCROLL ==> CSR

ISPF LIBRARY:
PROJECT ==> markispf
GROUP   ==> first   ==> second   ==> third   ==> fourth
TYPE    ==> source
MEMBER  ==> (Blank for member selection list)

OTHER PARTITIONED OR SEQUENTIAL DATASET:
DATASET NAME ==>
VOLUME SERIAL ==> (If not cataloged)

COMLIBS:
COMLIB      ==> first
COMLIB1     ==> second
COMLIB2     ==> third
COMLIB3     ==> fourth
ITEM NAME   ==> (Blank for member selection list)

DATASET PASSWORD ==> (If password protected)
EXCLUSIVE USE OF PRIMARY COMLIB ==> N (Y-Yes, N-No)

```

Figure 2-20 Concatenating Libraries on the Item Entry Panel

VISION:Workbench processes concatenated libraries in the same manner as ISPF. When using concatenated libraries, VISION:Workbench searches the libraries in order (library 1 through library 4), looking for the first occurrence of the item. If the item is not found, VISION:Workbench assumes that you are creating a new item.

Unless specifically overridden on the Save panel, VISION:Workbench for ISPF always writes the item out to the first or primary library in the specified Item Entry concatenation sequence. This is true even if the item was retrieved from another library.

## Library Considerations

The following sections discuss considerations relevant to each of the library types: partitioned data set (PDS), sequential data set, and COMLIB.

### Partitioned Data Set Considerations (PDS)

This section will discuss how to specify the name of the item that you want to edit.

[Figure 2-21](#) shows the two different areas on a VISION:Workbench for ISPF Item Entry panel that can be used to specify partitioned data sets.

The screenshot shows the 'ITEMENTRY' panel with the title 'DEFINITION - ITEM ENTRY PANEL'. It includes a 'COMMAND' field and a 'SCROLL' field. The main content is divided into two sections: 'ISPF LIBRARY' and 'OTHER PARTITIONED OR SEQUENTIAL DATASET'. The 'ISPF LIBRARY' section has fields for PROJECT, GROUP, TYPE, and MEMBER. The 'OTHER PARTITIONED OR SEQUENTIAL DATASET' section has fields for DATASET NAME, VOLUME SERIAL, COMLIBS, DATASET PASSWORD, and EXCLUSIVE USE OF PRIMARY COMLIB. A dashed box on the left highlights the 'ISPF LIBRARY' and 'OTHER PARTITIONED OR SEQUENTIAL DATASET' sections, with a note: 'Partitioned datasets can be specified in these areas.'

```

ITEMENTRY ----- DEFINITION - ITEM ENTRY PANEL -----
COMMAND ==>
SCROLL ==> CSR

> ISPF LIBRARY:
  PROJECT ==> MARKISPF
  GROUP   ==> ISPLAG1
  TYPE    ==> M4INPUT
  MEMBER  ==> (Blank for member selection list)

> OTHER PARTITIONED OR SEQUENTIAL DATASET:
  DATASET NAME ==> 'MARKISPF.ISPRTR1.SOURCE'
  VOLUME SERIAL ==> (If not cataloged)

COMLIBS:
  COMLIB      ==> 'ISPLAG1.M4LIB'
  COMLIB1     ==>
  COMLIB2     ==>
  COMLIB3     ==>
  ITEM NAME   ==> (Blank for member selection list)

DATASET PASSWORD ==> (If password protected)
EXCLUSIVE USE OF PRIMARY COMLIB ==> N (Y-Yes, N-No)
  
```

Figure 2-21 Specifying Partitioned Data Sets on the Item Entry Panel

If you use one of these areas to specify a Partitioned Data Set (PDS), then you must also supply a member name. There are two ways to provide a member name:

- The first way is to include the member name on the Item Entry panel.
- The second way is to select a member from the VISION:Workbench for ISPF member selection list. If you specify a partitioned data set and you do not supply a member name on the Item Entry panel, a member selection list is automatically displayed.

When specifying member names, VISION:Workbench for ISPF requires that the specified member name be identical to the member's item name. In other words, when editing an existing member from a partitioned data set, the specified member name must be identical to the item name that is specified in the first eight positions of the item's run control information. If these two names are not identical, VISION:Workbench for ISPF automatically changes the item name specified in the run control source to match the specified member name when the item is saved.

If you are creating a new item, you can choose any member name that does not already exist in the specified item entry source libraries. When the new item is saved, the specified member name is used as the item name in the generated run control information.

## Sequential Data Set Considerations

When specifying a sequential data set, you must provide the item name that is specified in the run control statements of the item being retrieved. If the item name that you provide does not match the item name specified in the run control statements, VISION:Workbench for ISPF automatically sets the run control information to the item name that you specified on the Item Entry panel (the 1-8 position name for the definition or application as known by VISION:Builder or VISION:Transact).

When creating a new item, you can choose any item name. Unless specifically overridden on the Save panel, the name that you choose is used to set the item name in the run control statements of the generated source.

Figure 2-22 shows how an item name is specified when using a sequential data set.

ITEMENTRY		DEFINITION - ITEM ENTRY PANEL	
COMMAND ==>		SCROLL ==> CSR	
ISPF LIBRARY:			
PROJECT	==> MARKISPF		
GROUP	==> ISPLAG1	==>	==>
TYPE	==> M4INPUT		
MEMBER	==>	(Blank for member selection list)	
OTHER PARTITIONED OR SEQUENTIAL DATASET:			
DATASET NAME	==> 'MARKISPF.ISPLCB1.FSINPUT(ITEMNAM)'		
VOLUME SERIAL	==>	(If not cataloged)	
COMLIBS:			
COMLIB	==> 'ISPLAG1.M4LIB'		
COMLIB1	==>		
COMLIB2	==>		
COMLIB3	==>		
ITEM NAME	==>	(Blank for member selection list)	
DATASET PASSWORD	==>	(If password protected)	
EXCLUSIVE USE OF PRIMARY COMLIB ==> N (Y-Yes, N-No)			

Figure 2-22 Using a Sequential Data Set

The item name is specified in the same way that the member name is specified for a partitioned data set.

## COMLIB Considerations

When specifying a COMLIB as your source library, you must also specify an item name. There are two ways to specify an item name.

- The first way is to include the item name on the Item Entry panel.
- The second way is to select an item from the VISION:Workbench for ISPF Item Selection List. If you specify a COMLIB and you do not supply an item name on the Item Entry panel, an Item Selection List is automatically displayed.

When specifying item names, VISION:Workbench for ISPF requires that the specified item name be identical to the item name that is specified in the item's run control information. If these two names are not identical VISION:Workbench for ISPF automatically sets the item name in the run control statements to match the item name specified on Item Entry panel or Selection List when the item is saved.

When creating a new item, you can choose any item name that does not already exist in the specified COMLIB(s). When the new item is saved, the name that you have specified is used to set the item name in the item's run control statements.

## Item Selection Lists

For simplicity, the term "Item Name" is used throughout this guide to refer to both member name entry (partitioned data set) and item name entry (sequential data set and COMLIBs).

If you have not specified a member name or item name, VISION:Workbench for ISPF displays the Item Selection List, shown in [Figure 2-23](#).

ITMSELECT---- ISPLCB1.MISPF.SOURCE -----										ROW 1 TO 12 OF 12	
COMMAND ==>										SCROLL ==> DATA	
	Name	Newname	LIB	UU.MM	Created	Last Modified	Size	Init	Mod	ID	
-	ACCOUNT		1	01 02	92/08/18	92/12/21 11:36	7	5	0	ISPLCB1	
-	CANCEL		1	01 00	93/05/26	93/05/26 09:23	3	3	0	ISPLCB1	
-	EMPFILE		1	01 00	93/05/26	93/05/26 09:53	11	11	0	ISPLCB1	
S	EMPLOYEE		1	01 01	92/12/21	92/12/21 13:54	3	3	0	ISPLCB1	
-	END		1	01 02	93/05/17	93/05/17 13:30	8	8	0	ISPLCB1	
-	FINANCE		1	01 13	89/05/05	92/12/21 11:52	171	108	0	ISPLCB1	
-	LESLIE		1	01 01	93/03/29	93/03/29 08:29	2	2	0	ISPLCB1	
-	LOGFD		1	01 06	89/05/04	92/09/02 17:31	182	154	0	ISPLAG1	
-	OLYMPICS		1	01 01	92/11/03	92/11/03 16:31	112	112	0	ISPLAG1	
-	QUARTER		1	01 01	92/08/18	92/08/18 15:35	5	5	0	ISPLAG1	
-	README		1	01 00	92/12/18	92/12/18 15:34	10	10	0	ISPLAG1	
***** BOTTOM OF DATA *****											

Figure 2-23 Using the Item Selection List

The VISION:Workbench for ISPF Item Selection List works in a manner very similar to the standard ISPF member selection list. To view this list, leave the item name blank on the Item Entry panel and press the Enter key.

From the Item Selection List, you can select an item by using the SELECT primary command or the Select line command.

The SELECT primary command is convenient for creating a new item, as shown in [Figure 2-24](#).

ITMSELECT---- ISPLCB1.MISPF.SOURCE -----										ROW 1 TO 12 OF 12
COMMAND ==> S NEWFILE										SCROLL ==> DATA
Name	Newname	LIB	UU.MM	Created	Last Modified	Size	Init	Mod	ID	
_ ACCOUNT		1	01 02	92/08/18	92/12/21 11:36	7	5	0	ISPLCB1	
_ CANCEL		1	01 00	93/05/26	93/05/26 09:23	3	3	0	ISPLCB1	
_ EMPFILE		1	01 00	93/05/26	93/05/26 09:53	11	11	0	ISPLCB1	
_ EMPLOYEE		1	01 01	92/12/21	92/12/21 13:54	3	3	0	ISPLCB1	
_ END		1	01 02	93/05/17	93/05/17 13:30	8	8	0	ISPLCB1	
_ FINANCE		1	01 13	89/05/05	92/12/21 11:52	171	108	0	ISPLCB1	
_ LESLIE		1	01 01	93/03/29	93/03/29 08:29	2	2	0	ISPLCB1	
_ LOGFD		1	01 06	89/05/04	92/09/02 17:31	182	154	0	ISPLAG1	
_ OLYMPICS		1	01 01	92/11/03	92/11/03 16:31	112	112	0	ISPLAG1	
_ QUARTER		1	01 01	92/08/18	92/08/18 15:35	5	5	0	ISPLAG1	
_ README		1	01 00	92/12/18	92/12/18 15:34	10	10	0	ISPLAG1	
***** BOTTOM OF DATA *****										

Figure 2-24 The SELECT Primary Command

The Select line command can be used to select an existing item. When using the Select line command there is also a New Name option. To take advantage of this feature, enter S, for Select, in the line command field and enter the new name under the New Name field. This allows you to create a copy of an existing item. This is demonstrated in [Figure 2-25](#).

ITMSELECT---- ISPLCB1.MISPF.SOURCE -----										ROW 1 TO 12 OF 12
COMMAND ==>										SCROLL ==> DATA
Name	Newname	LIB	UU.MM	Created	Last Modified	Size	Init	Mod	ID	
_ ACCOUNT		1	01 02	92/08/18	92/12/21 11:36	7	5	0	ISPLCB1	
_ CANCEL		1	01 00	93/05/26	93/05/26 09:23	3	3	0	ISPLCB1	
_ EMPFILE		1	01 00	93/05/26	93/05/26 09:53	11	11	0	ISPLCB1	
S EMPLOYEE	NEWFILE	1	01 01	92/12/21	92/12/21 13:54	3	3	0	ISPLCB1	
_ END		1	01 02	93/05/17	93/05/17 13:30	8	8	0	ISPLCB1	
_ FINANCE		1	01 13	89/05/05	92/12/21 11:52	171	108	0	ISPLCB1	
_ LESLIE		1	01 01	93/03/29	93/03/29 08:29	2	2	0	ISPLCB1	
_ LOGFD		1	01 06	89/05/04	92/09/02 17:31	182	154	0	ISPLAG1	
_ OLYMPICS		1	01 01	92/11/03	92/11/03 16:31	112	112	0	ISPLAG1	
_ QUARTER		1	01 01	92/08/18	92/08/18 15:35	5	5	0	ISPLAG1	
_ README		1	01 00	92/12/18	92/12/18 15:34	10	10	0	ISPLAG1	
***** BOTTOM OF DATA *****										

Figure 2-25 Creating a Copy of an Existing Item

To help you quickly scroll to a particular item, the standard ISPF LOCATE primary command can be used on any VISION:Workbench for ISPF Item Selection List.

Once you have completed the appropriate Item Entry panel by specifying your source data sets and item name, VISION:Workbench for ISPF can proceed with the import process. During the import process, VISION:Workbench for ISPF loads the specified item into VISION:Workbench for ISPF internal tables. These tables are used to display the item in a structured, hierarchical manner.

### Step 3.: Specify Validation Libraries (VISION:Builder only)

Step 3. of the VISION:Workbench for ISPF edit process only applies to VISION:Builder applications. This step is automatically bypassed for all other item types.

The validation library step allows you to concatenate up to four validation COMLIBs. If you specify validation libraries on this panel and enter a Y in the “Names Validation On?” entry, information from these libraries is used during your edit session to assist in interactive validation of your entries. This enables VISION:Workbench for ISPF to perform a more thorough validation during your edit session.

Keep in mind that if the Specify Validation Libraries panel is displayed and the entries are left blank, you cannot perform a global validation of your application. The term “global validation” is explained during the discussion on Step 5 of the edit process.

When a VISION:Builder application has been selected, the Specify Validation Libraries panel (VALLIBS), shown in [Figure 2-26](#) is displayed after the Item Entry panel. VISION:Workbench for ISPF automatically bypasses this step for all other item types.

```
VALLIBS ----- SPECIFY VALIDATION LIBRARIES -----
COMMAND ==>

Specify the Comlibs that should be used to validate the current item:

      COMLIB      ==>
      COMLIB1     ==>
      COMLIB2     ==>
      COMLIB3     ==>

Indicate whether referenced field names should be cross-checked with
file definitions. This option can be changed during the session
with the NAMES ON and NAMES OFF primary commands.

      NAMES VALIDATION ON? ==> Y    blank or N = NO
                                   Y = YES

Enter END      to Continue processing
Enter CANCEL   to Terminate processing of the current item
```

Figure 2-26 Specifying Validation COMLIBs

When you have completed this panel, enter the END command to continue processing.



## Step 4.: Edit the Selected Item

Step 4. of the edit process is where you create and/or modify your source item by entering information on the VISION:Workbench for ISPF data entry panels.

As discussed earlier, VISION:Workbench for ISPF data entry panels form a hierarchy. Using this hierarchy enables VISION:Workbench for ISPF to display an application or definition in a structured and organized manner. To step down through this hierarchy, use the S, for Select, in the line command field of the scrollable data row that you want to continue processing. If you select a row for which there are no more lower level panels, the following informational message is displayed:

COMMAND INACTIVE - COMMAND S, (SELECT), IS NOT VALID HERE.

To step back up through the hierarchy, use the END primary command (usually PF3).

Whenever you are uncertain about how to complete a data entry panel, remember to use the interactive Help facility. You can request Help for the entire screen using the HELP primary command or for a particular field by entering “?” in the first byte of the field in question.

### Field Name Validation (VISION:Builder only)

Field name validation is the process of searching the COMLIBs entered on the list of validation libraries for fields referenced in your application. If the referenced field is not found in the appropriate file or array definition in one of the validation libraries, an error message is issued. This type of validation is done for the following fields:

- Output field name in traditional (non-ASL) requests.
- Operands A, B, and C in traditional requests when the corresponding qualifier refers to one of the following file usages: M4NEW, M4OLD, M4TRAN, M4CORDn, ARRAYn, or ARRAYx.

There are three ways to turn off name validation:

1. You can enter an N in the “Names Validation On?” entry on the application validation panel. See [Figure 2-26](#).
2. The NAMES ON and NAMES OFF primary commands can be used to turn names validation on and off during the application edit session. These commands can be used as many times as needed to toggle the status of name validation during the edit session. The NAMES OFF command does not affect global validation (discussed in [Step 5.](#)).
3. While names validation is active, you can enter P, for Pass line command, to force VISION:Workbench for ISPF to recognize an undefined field name as valid for the duration of the session. However, if the application contains numerous undefined fields, it may be necessary to use the NAMES OFF command in order to continue the edit session.

Regardless of whether names validation is on or off, VISION:Workbench for ISPF always prevents you from entering invalid flag names in traditional (non-ASL) requests. (To get a list of valid VISION:Builder flags, enter the NAMES F primary command.)

### Duplicate Entry Checking (VISION:Builder only)

VISION:Workbench for ISPF also checks for invalid duplicate entries during an application edit session. Fields which are checked for invalid duplicate entries include array field names, field names within file definitions, request output numbers, request names, temporary field names, report sort sequences, file usages, and file segment names. Temporary field name validations do not take cataloged requests or ASL procedures into account when searching for duplicate names. However, these are taken into account during the global validation at the end of a VISION:Workbench for ISPF session.

You can prevent VISION:Workbench for ISPF from checking for invalid duplicate entries any time during your application edit session by using the DUP OFF primary command. The DUP ON primary command can be used to reactivate the duplicate checking. Each VISION:Workbench for ISPF session begins with duplicate checking ON.

## Step 5.: Validate the Item

Step 5. of the edit process does not apply to procedures, requests, and screens. If you are editing one of these item types, this step is automatically bypassed.

The validation step is invoked at the end of your edit session. This step gives you the opportunity to request a global validation of the item you are editing. The following describes how validation is performed for various items:

- For definition items, global validation is performed by invoking the COMLIB to validate the entire definition.
- For VISION:Builder applications, global validation is performed by invoking a VISION:Builder scan run.
- For VISION:Transact applications, global validation is performed by invoking an actual VISION:Transact APPGEN run.

During these validation runs, the entire item is checked not only for valid syntax, but for consistency among all associated entries across the entire item.

After the global validation has completed, you can browse the output generated during the validation processing. When browsing this listing, you can use the ISPF SCROLL RIGHT, LEFT, UP, and DOWN primary commands to examine the entire listing. You can also use the ISPF FIND primary command to look for specific character strings within the listing.

For example, [Figure 2-27](#) shows how you can use the ISPF FIND primary command to search for messages.

```

BROWSE -- BALMA07.M9TEMP1.LIST ----- Line 00000068 Col 001 080
COMMAND ==> F `TYPE 3'                SCROLL ==> PAGE
1MAR 27, 2002 13.05.30

*****
* REQUEST NAME - AROMCD1
* INPUT STREAM REQUEST
*****

-----
STMT REPORT                                MAX SEL SUM VERT FORMS PAGE PAGE L
TYPE DATE                                ITEMS CTL RPT SP  CNTRL WIDTH HGHT N
-----
(ER)

ST SUM V  8  PG  PG  I SP  MAXIMUM TB COLUMN DT  PG  STRT LIN SL RH .....S
TY RPT S LPI WID HGT M FRM LPP PGS TO HEADING POS POS PAGE NUM UB PN ENT   N
MP      P              G              LT TYP POS          NUM      ML TD REC

(E1)  (2)                                (N) (F) (B)  (MB)                                (ED)  (DOU
** MK4FR55  TYPE 3  AUTOMATIC FILE DEFINITION REQUESTED BUT NO CARD IMAGE SUBFI

-----
SP Q      E N S D C S T C C M M A M D Q PCT/
STMT SEQ BF L FIELD SEC- N P E E T U O U N A I V O E L FIE
TYPE NO. COL F NAME      LVL D R Q S L B T M T X N G D C F NAM
-----

```

Figure 2-27 Using the FIND Command to Search for Validation Errors

When you are done browsing the validation output, use the END primary command to continue.

## Validating Definitions

Because the actual validation process varies depending on the type of item you are editing, each type of validation is discussed separately.

The term “validation panel” is used when referring to general concepts. Specific validation panel titles are:

- Specify Validation Libraries panel (PDS, sequential data sets, and VISION:Builder)
- Specify Trial Compilation Libraries (VISION:Transact)

## Definitions Retrieved From COMLIB

When you edit a definition that was retrieved from a COMLIB, VISION:Workbench for ISPF assumes that upon completion of your edit session you want to re-catalog the modified item back into the primary COMLIB specified on the Item Entry panel. As a result, a global validation is automatically invoked to validate and re-catalog the definition. No validation panel is displayed.

If no errors are encountered during the validation process, the definition is automatically replaced back into the primary COMLIB. If errors are found, the definition is not replaced into the COMLIB and you are able to browse the output generated during the validation process. Remember, a definition that contains errors cannot be cataloged.

When you are done viewing the validation listing, you proceed to Step 6, Saving the Item.

### Definitions Retrieved From PDS or Sequential Data Set

If you are editing a file, array, logical data view, table, or transaction definition that was not retrieved from a COMLIB, the Specify Validation Libraries panel (VALDEF), shown in [Figure 2-28](#) is displayed at the end of your edit session.

```

VALDEF ----- SPECIFY VALIDATION LIBRARIES -----
COMMAND ==>

Specify the Comlibs that should be used to validate the current item:

      COMLIB      ==>
      COMLIB1     ==>
      COMLIB2     ==>
      COMLIB3     ==>

Catalog Valid Item? ==>      (Y - If valid, catalog item in first Comlib)
                               (N/Blank - Do Not Catalog item)

Press ENTER to Validate the current item
Enter END to By-Pass Validation processing and proceed to SaveAsis option
Enter CANCEL to Terminate Edit processing Without Saving the current item
  
```

Figure 2-28 Validation Panel for Items Not Retrieved from COMLIB

Validation panels allow you to specify validation libraries. A validation panel also allows you to specify whether or not the definition should be automatically cataloged. The definition is cataloged into the primary validation COMLIB, as well as being written back out to the primary source library specified on the Item Entry panel, if no errors are found.

If errors are encountered, you can browse the output generated during the validation process. When you are finished browsing the validation output, you automatically proceed to Step [6](#), Saving the Item.

## Validating VISION:Builder Applications

If you are editing a VISION:Builder application, the Specify Validation Libraries panel (VALAPP4), shown in [Figure 2-29](#) is displayed.

```
VALAPP4 ----- SPECIFY VALIDATION LIBRARIES -----  
COMMAND ==>  
  
If you proceed with Validation processing, the following Conlibs will be  
used to validate the current item:  
  
      COMLIB      ==>  
      COMLIB1     ==>  
      COMLIB2     ==>  
      COMLIB3     ==>  
  
Press ENTER to Validate the current item  
Enter END   to By-Pass Validation processing and proceed to SaveAsis option  
Enter CANCEL to Terminate Edit processing Without Saving the current item
```

Figure 2-29 Validation Panel for a VISION:Builder Application

The VISION:Builder application Specify Validation Libraries panel is an output only panel. When editing a VISION:Builder application, you specify your validation libraries at the start of your edit session, so this panel simply serves to remind you which libraries have already been specified.

If you decide to validate the application, a VISION:Builder scan run is automatically invoked. When the scan run has completed, you can browse the generated output. You are always shown the validation output, regardless of whether any errors were encountered.

When you are done viewing the validation listing, processing proceeds to Step 6, Saving the Item.

If no validation libraries are specified on the validation library panel at the start of your edit session, this step is bypassed and processing automatically proceeds to the Save panel.

## Validating VISION:Transact Applications

If you are editing a VISION:Transact application, the validation panel is called the Specify Trial Compilation Libraries panel. This panel is shown in [Figure 2-30](#).

```
VALAPP5 ----- SPECIFY TRIAL COMPILATION LIBRARIES -----  
COMMAND ==>  
  
Specify the Comlibs that should be used to perform a Trial Compilation:  
  
COMLIB ==>  
COMLIB1 ==>  
COMLIB2 ==>  
COMLIB3 ==>  
  
Trial Compilation Notes:  
- Catalog actions specified for instream Screen Definitions and  
  Requests will be performed as part of the Trial Compilation.  
  
- If this application contains DB2 File Definitions, enter 'END' to  
  bypass validation.  
  
Press ENTER to Perform a Trial Compilation  
Enter END to By-Pass Trial Compilation and proceed to SaveAsis option  
Enter CANCEL to Terminate Edit processing Without Saving the current item
```

Figure 2-30 Validation Panel for a VISION:Transact Application

If you choose to perform a trial compilation of your VISION:Transact application, an actual VISION:Transact APPGEN run is automatically invoked. You must specify the validation libraries that you want used during the APPGEN run on this panel.

When the validation completes, you are shown the output from the APPGEN run if errors were encountered. If no errors are found, processing continues directly to the Save panel.

When you have completed the validation process, processing automatically continues to Step [6](#), Saving the Item.

Keep in mind that because your VISION:Transact application validation is performed by actually invoking a VISION:Transact application generation run, any definitions contained within the application are automatically cataloged into the primary validation library as part of validation processing.

## Step 6.: Saving the Item

The final step in the edit process, Step 6., allows you to save the item being edited back to the original member that it was retrieved from or to a different member or even different library if you do not want to replace the original item.

The Specify “Save Asis” Library panel shown in [Figure 2-31](#) is used to save.

```

SAVEASIS ----- SPECIFY "SAVE ASIS" LIBRARY -----
COMMAND ==>

ISPF LIBRARY:
  PROJECT ==>
  GROUP   ==>
  TYPE    ==>
  MEMBER  ==>

OTHER PARTITIONED OR SEQUENTIAL DATASET:
  DATASET NAME ==>
  VOLUME SERIAL ==>          (If not cataloged)

DATASET PASSWORD ==>

Press ENTER to Save the item "ASIS", No validation is performed
Enter End   to Return to Edit without saving the item
Enter Cancel to Terminate Edit processing Without Saving the current item

```

Figure 2-31 Saving an Item

To save your source, simply enter the name of either a partitioned or sequential data set to which you would like to save the source. Remember that when saving an item back into a partitioned data set, you do not have to save to the same member. You can save the item to a new member by specifying a member name that does not already exist. If you specify an existing member name, that member is automatically replaced and the item name is changed to match the member name.

Once you have entered the library name, press the Enter key to “export” or save the item. If you would like to return to your current edit session without saving the item, use the END command.

**Note:** To terminate your current edit session, without saving any of your changes, use the CANCEL primary command.

Remember that save processing is always invoked at the end of your edit session. However, you can save the item being edited at any time during your edit session using the SAVE or SAVE ASIS primary commands.

- If you are editing an item that supports the global validation feature, the SAVE command takes you directly to the validation library panel.
- If you are editing an item that does not support the global validation feature, the SAVE command takes you directly to the Save panel.
- The SAVE ASIS command always takes you immediately to the Save panel, bypassing the global validation step. Entering SAVEASIS (without a space between the words) is improper syntax for this command and is processed as if you entered SAVE.

## Introduction to the Import Option

VISION:Workbench for ISPF includes an Import option that provides an automated interactive process to help in the conversion of external data definitions into the format used by VISION:Builder, VISION:Transact, and VISION:Inform. The task of building file definitions from existing definitions that are in other formats is made simple through the use of the Import option dialogs and the Quick Start utilities.

The Import option and the Quick Start utilities are provided as a starting point in the preparation of file definitions. After a conversion is performed, additional tailoring may be needed. This can be accomplished by using the VISION:Workbench for ISPF File option from a Main menu.

The Quick Start conversion utilities used within the Import dialogs are documented in detail in this guide. These utilities can also be executed in a batch processing mode (see the Quick Start Reference Guide).

The Import option dialogs assume that you are conceptually familiar with the utilities and the control statements, and that you have read the chapters in the Quick Start guide that describe each utility.



When the Import option is selected from a VISION:Workbench for ISPF Main menu, the panel shown in [Figure 2-32](#) is displayed.

```

M9IM00  ----- IMPORT EXTERNAL DEFINITIONS -----
COMMAND ==> _

Select the Definition Type or processing option.

Enter the number on the command line and press ENTER.

          1 - COBOL Definitions
          2 - DB2 Table Definitions
          3 - VISION:Results Definitions
          4 - VISION:Inquiry Definitions
          5 - VISION:Builder Definitions

          9 - Duplicate Field Name Check

Use END or CANCEL to exit this process.

```

Figure 2-32 Import External Definitions Panel

There are several selections available to import and convert external definitions. Once selected, each subsequent dialog walks you through the process of entering information and executing the appropriate conversion utility.

The Import options execute the Quick Start utilities in foreground without having to prepare the JCL and control statements for batch job submission. The Import conversions are performed in foreground using dialog panels, screen information displays, and Quick Start utility programs. The panels prompt you for the information needed to run each utility. The information is edited and saved, and is available from session to session.

After the utilities are invoked, their output listing is displayed for review. The skeleton definitions are then checked for duplicate field names. You are returned to the Import Options Menu when the Import process is complete.

During the Import process, information screens are displayed to indicate the various foreground activities being performed, the results of the activities, and if any interaction is required. Messages are displayed to describe any conditions that may need attention. There are Help panels available throughout the dialogs to aid you in understanding the process and the specific entries on the various panels.

Each Import dialog is unique to the definition being converted and only builds one file definition at a time. There are similarities among the dialogs; however, the detail information entered for each process is maintained separately.

The use of function keys and commands is standardized within the Import option. Each panel indicates the action and result from an available function key or command. The following table is a general description of the function keys and commands available within the Import dialogs. The standard ISPF scrolling keys and commands are also available.

Key or Command	Description
Enter	Captures the input data and/or moves forward through the dialog process. When the data or information is captured, it is edited and then stored. The next dialog panel, when appropriate, is displayed after the Enter key is pressed. Any situation that prevents movement, like data errors or missing information, is indicated by a message displayed on the current panel.
End	Saves the data and/or list information and exits to the previous display. In some cases, the data on the current panel can be edited before exiting the display.
Cancel	Exits to the previous display without saving or validating the entered data or the information in a list.
Clear	Erases entries from the segment list so that a new list can be built. This causes the cleared segment list to be saved, replacing the previous information.
Run	Saves the entered data or information list and starts the execution of the appropriate Quick Start utility program.
Help	Displays help information associated with your current location within the dialog process.
Asis	Accepts the information as is with the understanding that further action may be needed for the information to be valid.

## COBOL Definitions

COBOL definitions (copybooks) are imported using the COBOL Quick Start utility, which produces a file definition from COBOL data definitions. After the skeleton file definition is built, it is automatically checked for duplicate field names. You are prompted for field name tailoring if duplicates are found. If additional tailoring is needed, use the VISION:Workbench for ISPF File option from a Main menu.

The COBOL definition copybooks can be retrieved from OS/390 (MVS) partitioned data sets, CA-Panvalet libraries, and CA-Librarian libraries.

**Note:** Additional installation preparations must be performed to setup the COBOL Quick Start utility for retrieval from the CA-Panvalet and CA-Librarian libraries. (See the Installation and Support Guide for details.)

The first panel in the COBOL definitions dialog process requests the general information needed for execution of the Quick Start utility. [Figure 2-33](#) shows an example of this initial panel.

```

M9IM10 ----- COBOL Quick Start Utility Run -----
COMMAND ==> _

Provide the General Information for the COBOL Quick Start Utility run.

Definition Library      ==> 'DEFS.LIBRARY'
Listing Data Set Name  ==> 'IMPORT.LISTING'

File Definition Name    ==> EMPFILE      (Member Name in the Definition Library)
File Type              ==> VARIABLE     (Leave blank for Pop-up Choice List)
Buffer Size            ==> 12K          (optional)
Field Name Prefix      ==> EMP          (optional)
Record Size            ==>              (optional - applies to Fixed/Variable)
Records per Block      ==>              (optional - applies to Fixed)

Press the ENTER key   to Capture the General Information
                        and Proceed to the Segment information Panel.

Use END               to Save and Exit this process.
Use CANCEL             to Exit this process without save.

```

Figure 2-33 COBOL Quick Start Utility Run — General Information Panel

The following is a description of the entries on the panel in [Figure 2-33](#).

Definition Library	Required. This is the source definition library in which the generated file definition will be written as output. This data set must be a PDS, fixed record format, and have an LRECL of 80.
Listing Data Set Name	Required. This is where the utility program will write an output report of information regarding the run. This data set is presented in browse mode at the end of the run so that you can review the utility information and any messages. This must be a sequential data set with a RECFM of F or FB and an LRECL of 133.

**File Definition Name**      Required. This is the file name used to identify the generated definition. This name will also be the member name of the item written to the definition library. The name must start with an alphabetic letter and can be up to eight alphanumeric characters in length.

If the member name already exists in the definition library, it will be replaced.

**File Type**                      Required. This indicates the file format type code for the file definition that is being generated in the run. [Figure 2-34](#) shows the pop-up window that is displayed when this entry is left blank.

```
M9IM10 ----- COBOL Quick Start Utility Run -----
COMMAND ==>

Provide the General Information for the COBOL Quick Start Utility run.

Definition Library    ==> DEFS.LIBRARY
Listing Data Set Name ==> IMPORT.LISTING

File Definition Name  ==> EMPFILE      (Member Name in the Definition Library)
File Type             ==>
Buffer Size           ==> 12K
Field Name Prefix     ==> EMP
Record Size           ==>
Records per Block     ==>

Press the ENTER key  to Capture the
                    and Proceed to

Use END   to Save and Exit this pro
Use CANCEL to Exit this process with

-----
M9IM11 -- FILE TYPE CHOICE WINDOW --
COMMAND ==> _
Press ENTER to process selection.

1. VSAM KSDS      7. Variable
2. VSAM ESDS      8. Fixed
3. VSAM AIX       9. Undefined
4. DB2            10. GDBI
5. DLI            11. ISAM Fixed
6. DLI HDAM       12. ISAM Variable
-----
```

Figure 2-34      COBOL Quick Start Utility Run — File Type Choice Pop-up Window

The pop-up window allows you to select the file type value by entering the appropriate number. The file type entry value can be keyed direct by using the associated values as shown in the list below. The following table is a description of the file types available.

File Type	Description
DB2	DB2 Relational Database
KSDS	VSAM Key Sequenced Data Set
ESDS	VSAM Entry Sequenced Data Set
AIX	VSAM Alternate Index Data Set
DLI	IMS Database
DLIHDAM	IMS HDAM Database
ISAMFIX	ISAM Fixed Length Record Format Data Set
ISAMVAR	ISAM Variable Length Record Format Data Set
FIXED	Fixed Length Record Format Data Set
VARIABLE	Variable Length Record Format Data Set
UNDEFINED	Undefined Record Format Data Set
GDBI	Generalized Data Base Interface Mapped File

Buffer Size	Optional. This is the maximum record size for database file types. For the simple fixed and variable file types, this is the block size. Use a number from 1 to 32760 or value of 1K to 9999K.
Field Name Prefix	Optional. This is a 1 to 3-character prefix that is used for the primary field names of the generated file definition. Because imported field names can be longer than 8 characters, primary field names are automatically generated using this prefix and a sequential number. The prefix must start with an alphabetic letter. The letter F will be used if no prefix is specified.
Record Size	Optional. This can be specified for the simple fixed and variable file types by entering a number from 1 to 9999. For database file types, this entry should be left blank.
Records per Block	Optional. This can be specified for the fixed file types by entering a number from 1 to 999. For database and variable file types, this entry should be left blank.

Once the general information for the COBOL Quick Start utility run is captured, the next panel is displayed to capture the detail information for the segments and file structure.

The information for the segments within the file definition is captured in a segment list. The list is built and manipulated by entering data and actions on the input line. The list is ordered and referenced by the segment numbers. The information entered is used to define the file segment structure, name the segments, and identify the source of the data definition. [Figure 2-35](#) shows the panel used to enter the segment information.

```
M9IM12 ----- COBOL Quick Start Utility Run ----- ROW 1 TO 3 OF 3
COMMAND ==> _

      (A) Add , (C) Change or (D) Delete Segment Information.

      Action Number/Level/Name      BookName/Type (Mvs,Pan,Libr)
Input ==> .      ..      .      .....      .....

Press ENTER to capture the Information into the Segment List below.
Use CLEAR Command to ERASE all entries from the Segment List below.
Use END to SAVE the List and Exit.
Use RUN to SAVE the List and Execute the Utility program.
Use CANCEL to exit without save.

      File Definition EMPFILE      Buffer Size 12K      Field Prefix EMP
The Segment List --- Number/Level/Name      BookName/Type (Mvs,Pan,Libr)
                        10      1 OFFSEG      JKOFFICE M
                        20      2 DEPTSEG      JKDEPT P
                        30      3 EMPSEG      JKEMP L
***** BOTTOM OF DATA *****
```

Figure 2-35 COBOL Quick Start Utility Run — Segment Information Panel

The Input line entries are as follows:

- Action

A Add an item to the segment list.  
C Change the information for an item in the segment list.  
D Delete an item from the segment list.

The Clear primary command can be used to remove all of the information from the segment list.
- Number

Use a number from 1 to 99 to identify the segment and the location within the file structure. Subordinate segments will have a higher number, while parent segments will have a lower number. Segment numbering is top to bottom, left to right within the structure.

Level	Use a number from 1 to 9 to identify the subordination of the segment within the file structure. The root segment must have a level of 1. All subordinate segments will have a number from 2 to 9.
Name	Enter a segment name starting with an alphabetic letter followed by alphanumeric characters. The name can be from 1 to 8 characters long.
Bookname	Enter the name of the copybook that contains the field definitions to be used when generating this segment definition.
Type (MVS,PAN, LIBR)	Enter a letter, M, P, L, that indicates the source library data set type where the copybook is stored. M identifies an OS/390 (MVS) PDS data set. P identifies a CA-Panvalet library. L identifies a CA-Librarian library.

After the segment information has been entered, the utility is ready to run.

To start the utility, enter the RUN primary command. In order to execute the COBOL Quick Start utility, the data set names of the libraries that contain the data definition copybooks associated with the segment information are needed. Only the names of the data set types actually referenced in the segment list information need to be specified. [Figure 2-36](#) shows the panel displayed to capture the copybook data set information.

```

M9IM13 ----- COBOL Quick Start Utility Run -----
COMMAND ==> _

Provide the CopyBook Library Names for the COBOL Quick Start Utility run.

                                Copybook Data Set Names .....

MVS      Library      ==> MVS.LIB
Panvalet Library      ==> PAN.LIB
Librarian Library     ==> LBR.LIB

(Note that only the libraries being used need to be supplied.)

Press the ENTER key to Capture the Information and continue.

Use END    to Save and Exit this process.
Use CANCEL to Exit this process without save.

```

Figure 2-36 COBOL Quick Start Utility Run — Copybook Library Names Panel

The segment information is scanned and you are prompted for the required data set names if they are not entered. The panel entries are as follows.

OS/390 (MVS) Library	Data set name indicates a PDS for copybook retrieval.
Panvalet Library	Data set name indicates a CA-Panvalet library for copybook retrieval.
Librarian Library	Data set name indicates a CA-Librarian library for copybook retrieval.

Once the data set names are captured, the utility starts executing. The screen will contain information indicating that the utility is running. Text will be displayed to indicate that the utility has completed and you can browse the output listing. When the output browse is ended, the dialog process automatically performs a duplicate field name check. The screen will contain text indicating that the checks are being performed. When the checks are completed, the screen will contain text indicating the results of the checks. If duplicate names are found, you will be directed to a panel that lets you review and change the field names. See [Editing Items on page 2-21](#) for details on the Duplicate Field Name Check option.

The dialog completes by returning to the Import Options Main menu.

## DB2 Table Definitions

DB2 table definitions are imported using the DB2 Quick Start utility, which produces a file definition from DB2 table definitions in the DB2 system catalog. After the skeleton file definition is built, it is checked for duplicate field names. You are prompted for field name tailoring if duplicates are found. If additional tailoring is needed, use the VISION:Workbench for ISPF File option from a Main menu.

**Note:** A DB2 bind, using a supplied DBRM, to create a DB2 plan must be performed before the utility can be executed. (See the Installation and Support Guide for details.)



The first panel requests the general information needed for execution of the DB2 Quick Start utility. [Figure 2-37](#) shows an example of the initial panel.

```

M9IM20 ----- DB2 Quick Start Utility Run -----
COMMAND ===> _

Provide the General Information for the DB2 Quick Start Utility run.

Definition Library      ===> 'DEFS.LIBRARY'
Listing Data Set Name   ===> 'IMPORT.LISTING'

DB2 Plan Name          ===> INM4JJK2
DB2 Subsystem ID       ===> DB2T

File Definition Name    ===> CUSTFILE  (Member Name in the Definition Library)
Buffer Size            ===> 50K       (optional)
Field Name Prefix       ===> CS       (optional)

Press the ENTER key to Capture the General Information and
                        and Proceed to the Segment information Panel.

Use END to Save and Exit this process.
Use CANCEL to Exit this process without save.

```

Figure 2-37 DB2 Quick Start Utility Run — General Information Panel

The following is a description of the entries on the panel in [Figure 2-37](#).

Definition Library	Required. This is the source definition library in which the generated file definition will be written as output. This data set must be a PDS, fixed record format, and have an LRECL of 80.
Listing Data Set Name	Required. This is where the utility program will write an output report of information regarding the run. This data set is presented in browse mode at the end of the run so that you can review the utility information and any messages. This must be a sequential data set with a RECFM of F or FB and an LRECL of 133.
DB2 Plan Name	Required. This is the plan name used during the installation process when binding the DB2 Quick Start utility. The TSO Call Attach facility is used to connect with DB2. Static SQL is used to process information from the DB2 SYSCOLUMNS table.
DB2 Subsystem ID	Required. This identifies the DB2 system that will be accessed for the table definition information.

File Definition Name	<p>Required. This is the file name used to identify the generated definition. This name will also be the member name of the item written to the definition library. The name must start with an alphabetic letter and can be up to eight alphanumeric characters in length.</p> <p>If the member name already exists in the definition library, it will be replaced.</p>
Buffer Size	<p>Optional. This defines the size of the buffer needed to hold the maximum logical record. Use a number from 1 to 32760 or value of 1K to 9999K.</p>
Field Name Prefix	<p>Optional. This is a 1 to 3-character prefix that is used for the primary field names of the generated file definition. Because the DB2 column names can be longer than 8 characters, primary field names are automatically generated using this prefix and a sequential number. The prefix must start with an alphabetic letter. The letter F will be used if no prefix is specified.</p>

Once the general information for the DB2 Quick Start utility run is captured, the next panel is displayed to capture the detail information for the segments and file structure.

The information for the segments within the file definition is captured in a segment list. The list is built and manipulated by entering data and actions on the input line. The list is ordered and referenced by the segment numbers. The information entered is used to define the file segment structure, name the segments, and identify the source of the data definition. [Figure 2-38](#) shows the panel used to enter the segment information.

```

M9IM21 ----- DB2 Quick Start Utility Run ----- ROW 1 TO 6 OF 6
COMMAND ==> _

      (A) Add , (C) Change or (D) Delete Segment Information.

      Action Number/Level/Name      DB2 Table Name and Creator
Input ==> .      ..      .      .....

Press ENTER to capture the Information into the Segment List below.
Use CLEAR Command to ERASE all entries from the Segment List below.
Use END to SAVE the List and Exit.
Use RUN to SAVE the List and Execute the Utility program.
Use CANCEL to exit without save.

      File Definition CUSTFILE Buffer Size 50K Field Prefix CS
The Segment List --- Number/Level/Name      DB2 Table Name and Creator
                        10      1 CUSTSEG      JJKS10      ISPJK1
                        20      2 ADDRSEG      JJKS20      ISPJK1
                        30      2 SITESEG      JJKS30      ISPJK1
                        40      3 PRODSEG      JJKS40      ISPJK1
                        50      4 SYSSEG      JJKS50      ISPJK1
                        60      4 ENVRSEG      JJKS60      ISPJK1
***** BOTTOM OF DATA *****

```

Figure 2-38 DB2 Quick Start Utility Run — Segment Information Panel

The Input line entries are as follows:

Action	A Add an item to the list.
	C Change the information for an item in the list.
	D Delete an item from the list.

The Clear primary command can be used to remove all of the information from the segment list.

Number	Use a number from 1 to 99 to identify the segment and the location within the file structure. Subordinate segments will have a higher number, while parent segments will have a lower number. Segment numbering is top to bottom, left to right within the structure.
--------	---

Level	Use a number from 1 to 9 to identify the subordination of the segment within the file structure. The root segment must have a level of 1. All subordinate segments will have a number from 2 to 9.
-------	--

Name	Enter a segment name starting with an alphabetic letter followed by alphanumeric characters. The name can be from 1 to 8 characters long.
------	---

DB2 Table Name	Enter the name of the DB2 table that contains the column information to be used when generating the segment field definitions.
Creator	Enter the authorization ID, creator ID, or an asterisk (*) to qualify the DB2 table name being accessed for field information. An asterisk (*) retrieves field information from all tables with the specified name.

After the segment information has been entered, the utility is ready to run.

The utility will access the DB2 SYSCOLUMNS table and extract the column information for referenced tables in the segment list information.

Once the segment information is captured, enter the RUN primary command to start executing the utility. The screen will contain information indicating that the utility is running. Text will be displayed to indicate that the utility has completed and you can browse the output listing. When the output browse is ended, the dialog process automatically performs a duplicate field name check. The screen will contain text indicating that the checks are being performed. When the checks are completed, the screen will contain text indicating the results of the checks. If duplicate names are found, you will be directed to a panel that lets you review and change the field names. See [Editing Items on page 2-21](#) for details on the Duplicate Field Name Check option.

The dialog completes by returning to the Import Options Main menu.

## VISION:Results Definitions

VISION:Results™ definitions are imported using the VISION:Results Quick Start utility, which produces a file definition from a VISION:Results file definition. After the skeleton file definition is built, it is checked for duplicate field names. You are prompted for field name tailoring if duplicates are found. If additional tailoring is needed, use the VISION:Workbench for ISPF File option from a Main menu.

The VISION:Results definitions can be retrieved from OS/390 (MVS) Partitioned Data Sets, CA-Panvalet libraries, and CA-Librarian libraries.

**Note:** Additional installation preparations must be performed to setup the VISION:Results Quick Start utility for retrieval from the CA-Panvalet and CA-Librarian libraries. (See the Installation and Support Guide for details.)

A panel is displayed to gather all the information needed to execute the Quick Start utility. [Figure 2-39](#) shows an example of the panel.

```

M9IM30 ----- VISION:Results Quick Start Utility Run -----
COMMAND ===> _

Provide the Information for the VISION:Results Quick Start Utility run.

Definition Library    ===> 'DEFS.LIBRARY'
Listing Data Set Name ===> 'IMPORT.LISTING'

Results Definition    ===> RSACCTS  (The Results File Name will become the)
                                   (Member Name in the Definition Library)

Copybook Library Type ===> P        (M,P,L - MVS, Panvalet, Librarian)
MVS PDS Library      ===>
Panvalet Library      ===> 'ISPJK1.PAN.LIB'
Librarian Library     ===>

(Note that only the library being used needs to be supplied.)

Press the ENTER key to Capture the Information

Use END   to Save and Exit this process.
Use RUN   to Save and Execute the Utility program.
Use CANCEL to Exit this process without save.

```

Figure 2-39 VISION:Results Quick Start Utility Run — Information Panel

The following is a description of the entries on the panel in [Figure 2-39](#).

Definition Library	Required. This is the source definition library in which the generated file definition will be written as output. This data set must be a PDS, fixed record format, and have an LRECL of 80.
Listing Data Set Name	Required. This is where the utility program will write an output report of information regarding the run. This data set is presented in browse mode at the end of the run so that you can review the utility information and any messages. This must be a sequential data set with a RECFM of F or FB and an LRECL of 133.

Results Definition	<p>Required. This is the name of the VISION:Results definition and will identify the generated definition. This name will also be the member name of the item written to the definition library. The name must start with an alphabetic letter and can be up to eight alphanumeric characters in length.</p> <p>If the member name already exists in the definition library, it will be replaced.</p>
Copybook Library Type	<p>Enter a letter, M, P, L, to indicate the source library data set type where the Results definition is stored.</p> <p>M indicates that the copybook is stored in a OS/390 (MVS) PDS data set.</p> <p>P indicates that the copybook is stored in a CA-Panvalet library.</p> <p>L indicates that the copybook is stored in a CA-Librarian library.</p> <p>The data set name that corresponds to the value entered here must be supplied in one of the subsequent entries on this panel.</p>
OS/390 (MVS) PDS Library	<p>The OS/390 (MVS) data set name where the VISION:Results definition is stored for copybook retrieval.</p>
Panvalet Library	<p>The CA-Panvalet data set name where the VISION:Results definition is stored for copybook retrieval.</p>
Librarian Library	<p>The CA-Librarian data set name where the VISION:Results definition is stored for copybook retrieval.</p>

The information on the panel is scanned and you are prompted for the required copybook data set names if they are not entered.

Once the information on this panel is captured, enter the RUN primary command to start executing the utility. The screen will contain information indicating that the utility is running. Text will be displayed to indicate that the utility has completed and you can browse the output listing. When the output browse is ended, the dialog process automatically performs a duplicate field name check. The screen will contain text indicating that the checks are being performed. When the checks are completed, the screen will contain text indicating the results of the checks. If duplicate names are found, you will be directed to a panel that lets you review and change the field names. See [Editing Items on page 2-21](#) for details on the Duplicate Field Name Check option.

The dialog completes by returning to the Import Option menu.

## VISION:Inquiry Definitions

VISION:Inquiry<sup>®</sup> definitions are imported using the VISION:Inquiry Quick Start utility, which produces a file definition from VISION:Inquiry database maps. After the skeleton file definition is built, it is checked for duplicate field names. You are prompted for field name tailoring if duplicates are found. If additional tailoring is needed, use the VISION:Workbench for ISPF File option from a Main menu.

The VISION:Inquiry database maps are retrieved from the unloaded sequential copy of the system database created by the VISION:Inquiry IXULOAD utility. The IXULOAD utility is described in the VISION:Inquiry Technical Reference Guide.

A panel is displayed to gather all the information needed to execute the Quick Start utility. [Figure 2-40](#) shows an example of the panel.

```

M9IM40 ----- VISION:Inquiry Quick Start Utility Run -----
COMMAND ===> _

Provide the Information for the VISION:Inquiry Quick Start Utility run.

Definition Library      ===> 'DEFS.LIBRARY'
Listing Data Set Name  ===> 'IMPORT.LISTING'

UNLOADED Inquiry
System Data Base       ===> 'IMS60.IXXDB.UNLOAD'

Inquiry Definition     ===> VSHPLANT (This name or Rename will become the )
                                (Member Name in the Definition Library)
Rename the Definition  ===> PLANTFIL (optional)

Buffer Size            ===> 15000 (optional)
Field Name Prefix      ===> PL (optional)

Press the ENTER key to Capture the Information

Use END   to Save and Exit this process.
Use RUN   to Save and Execute the Utility program.
Use CANCEL to Exit this process without save.

```

Figure 2-40 VISION:Inquiry Quick Start Utility Run — Information Panel

The following is a description of the entries on the panel in [Figure 2-40](#).

Definition Library	Required. This is the source definition library in which the generated file definition will be written as output. This data set must be a PDS, fixed record format, and have an LRECL of 80.
Listing Data Set Name	Required. This is where the utility program will write an output report of information regarding the run. This data set is presented in browse mode at the end of the run so that you can review the utility information and any messages. This must be a sequential data set with a RECFM of F or FB and an LRECL of 133.
UNLOADED Inquiry System Data Base	Required. This is the name of the sequential data set that contains the unloaded system database created by the VISION:Inquiry IXULOAD utility.



Inquiry Definition	<p>Required. This is the name of the database map to be converted. This name (or the Rename entry) will be used to identify the generated definition. This name will also be the member name of the item written to the definition library. The name must start with an alphabetic letter and can be up to eight alphanumeric characters in length.</p> <p>If the member name already exists in the definition library, it will be replaced.</p>
Rename the Definition	<p>Optional. This is used to change the database map name on the generated definition. If used, this name will be used to identify the generated definition. This name will also be the member name of the item written to the definition library. The name must start with an alphabetic letter and can be up to eight alphanumeric characters in length.</p> <p>If the member name already exists in the definition library, it will be replaced.</p>
Buffer Size	<p>Optional. This defines the size of the buffer needed to hold the maximum logical record. Use a number from 1 to 32760 or value of 1K to 9999K. This value is only used if the imported definition is for an IMS database.</p>
Field Name Prefix	<p>Optional. This is a 1 to 3-character prefix that is used for the primary field names of the generated file definition. Because imported field names can be longer than 8 characters, primary field names are automatically generated using this prefix and a sequential number. The prefix must start with an alphabetic letter. The letter F will be used if no prefix is specified.</p>

Once the information is captured, enter the RUN primary command to start executing the utility. The screen will contain information indicating that the utility is running. Text will be displayed to indicate that the utility has completed and you can browse the output listing. When the output browse is ended, the dialog process automatically performs a duplicate field name check. The screen will contain text indicating that the checks are being performed. When the checks are completed, the screen will contain text indicating the results of the checks. If duplicate names are found, you will be directed to a panel that lets you review and change the field names. See [Editing Items on page 2-21](#) for details on the Duplicate Field Name Check option.

The dialog completes by returning to the Import Option menu.

## VISION:Builder Definitions

VISION:Builder definitions are imported using a VISION:Builder source statement retrieval run that is identified here as the VISION:Builder Quick Start utility. This utility retrieves the file definition from a COMLIB common library. The file definition will already be in the correct format. There is no duplicate field name check performed because the definition will not contain duplicate field names. There is no need for additional tailoring. However, if any modifications are needed, such as alternate names or field descriptions, use the VISION:Workbench for ISPF File option from a Main menu.

The file definitions are retrieved from the COMLIB common library data set (known as M4LIBs). The access method format of the common library can be either BDAM or VSAM.

A panel is displayed to gather all the information needed to execute the Quick Start utility. [Figure 2-41](#) shows an example of the panel.

```

M9IM50 ----- VISION:Builder Quick Start Utility Run -----
COMMAND ===> _

Provide the Information for the VISION:Builder Quick Start Utility run.

Definition Library      ===> 'DEFS.LIBRARY'
Listing Data Set Name  ===> 'IMPORT.LISTING'

COMLIB Library Name    ===> 'JJKVSAM.COMLIB'

File Definition Name    ===> SALEAMTS  (File Definition name in COMLIB)

New Definition Name     ===> SALESFIL  (Leave blank to retain same name as)
                                         (Member Name in the Definition Library)

Press the ENTER key to Capture the Information

Use END    to Save and Exit this process.
Use RUN    to Save and Execute the Utility program.
Use CANCEL to Exit this process without save.

```

Figure 2-41 VISION:Builder Quick Start Utility Run — Information Panel

The following is a description of the entries on the panel in [Figure 2-41](#).

Definition Library	Required. This is the source definition library in which the generated file definition will be written as output. This data set must be a PDS, fixed record format, and have an LRECL of 80.
--------------------	--

Listing Data Set Name	Required. This is where the utility program will write an output report of information regarding the run. This data set is presented in browse mode at the end of the run so that you can review the utility information and any messages. This must be a sequential data set with a RECFM of F or FB and an LRECL of 133.
COMLIB Library Name	Required. This is the name of the common library data set that contains the file definitions to be retrieved.
File Definition Name	<p>Required. This is the file definition name of the item stored in the common library. This name will be used to identify the retrieved definition. This name will also be the member name of the item written to the definition library unless a new name is specified.</p> <p>If the member name already exists in the definition library, it will be replaced.</p>
New Definition Name	<p>Optional. This is used to change the file definition name of the retrieved item. The name must start with an alphabetic letter and can be up to eight alphanumeric characters in length. This name will also be the member name of the item written to the definition library.</p> <p>If the member name already exists in the definition library, it will be replaced.</p>

Once the information is captured, enter the RUN primary command to start executing the utility. The screen will contain information indicating that the utility is running. Text will be displayed to indicate that the utility has completed and you can browse the output listing. There is no duplicate field name check performed.

The dialog completes by returning to the Import Option menu.

## Duplicate Field Name Check

The Duplicate Field Name Check is performed automatically from the individual Import dialogs. This checking can also be performed on request against any file definition that was either generated, developed, or copied to a definition library.

The file definition will be retrieved and scanned. The primary field names and the alternate field names are checked for duplicate occurrences. Each field name in the definition must be unique. If duplicates are found, a panel will be displayed and changes can be entered. Once the changes are complete, the updated definition is rewritten to the definition library. The duplicate checking is performed in foreground using a procedure language program.

A panel is displayed to gather all the information needed to perform the duplicate field name check. [Figure 2-42](#) shows an example of the panel.

```
M9IM90 ----- Duplicate Field Name Check -----
COMMAND ===> _

Provide the Definition Library and Member Name.

Definition Library    ===> 'DEFS.LIBRARY'
File Definition Name  ===> EMPFILE

Press the ENTER key to Check for Duplicate Field Names.

Use END    to Save and Exit this process.
Use CANCEL to Exit this process without save.
```

Figure 2-42 Duplicate Field Name Check Panel

The following is a description of the entries on the panel.

Definition Library	Required. This is the source definition library that contains the file definition to be scanned for duplicates. This data set must be a PDS, fixed record format, and have an LRECL of 80.
File Definition Name	Required. This is the name (and PDS member name) of the file definition to be scanned for duplicates.  When the definition is changed and saved, it replaces the previous data.

Once the information is entered, press the ENTER key to start executing the duplicate field name program. The screen will contain text indicating that the checks are being performed. When the checks are completed, the screen will contain text indicating the results of the checks. If duplicate names are found, you are directed to a panel that lets you review and change the field names. (For more information, see [Duplicate Field Name Editor on page 2-59](#).)

## Duplicate Field Name Editor

The Duplicate Field Name Editor panel is displayed when duplicate fields names are found within a file definition generated by one of the individual Import dialogs. This panel process is shared by all functions within the Import option.

The Duplicate Field Name Editor panel contains a list of the primary and alternate field names from a file definition. The names are displayed in alphabetic order with the duplicate names flagged for correction. When the changes have been completed and the interaction ended, the file definition is checked again for duplicates. Only when no more duplicates are found, will the interaction return to the Import Option menu. [Figure 2-43](#) shows an example of panel.

```

M91MA00 ----- Duplicate Field Name Editor ----- ROW 63 TO 72 OF 72
COMMAND ==>

Duplicate field names for file definition EMPFILE  flagged for correction.

Press ENTER to process the name changes.
Use  END    to SAVE and Re-Check for Duplicate Names.
Use  CANCEL to Exit. (Changes are NOT Saved)
Use  ASIS   to SAVE ASIS and Exit.
Use  FIND   nnnnn... or * - go to a field name or first duplicate name

Duplicate Found      Primary/Alternate      Name      From Segment
                        Field Names          Type      Number/Name
                        OFFICE_DATA          ALT       10  OFFICE
                        OFFICE_PHONE          ALT       10  OFFICE
                        OFFICE_STATE          ALT       10  OFFICE
                        OFFICE_STREET         ALT       10  OFFICE
                        OFFICE_ZIP            ALT       10  OFFICE
                        OFFICE_ZIP_FIRST_FIVE ALT       10  OFFICE
                        OFFICE_ZIP_LAST_FOUR  ALT       10  OFFICE
                        SPEED_DIAL            ALT       10  OFFICE
                        * SPEED_DIAL          ALT       20  DEPTMT
                        * SPEED_DIAL          ALT       30  EMPLOYEE
***** BOTTOM OF DATA *****

```

Figure 2-43 Duplicate Field Name Editor Panel

The following is a description of the information the panel in [Figure 2-43](#).

Duplicate Found	This column contains an asterisk (*) for the field names that are duplicates as of the last duplicate check scan.
Primary/Alternate Field Names	The names are displayed in alphabetic order and can be changed here by typing over the name entry.
Name Type	This column indicates which type of name is displayed. Primary names are indicated by PRI, and alternate names are indicated by ALT.
From Segment Number/Name	This column identifies the segment within the file structure that contains the field name.

When this dialog completes, it displays text indicating whether duplicates were still found or if they have all been corrected. Then it returns to the Import Option menu.

## Introduction to Job Submission

VISION:Workbench for ISPF provides both VISION:Builder and VISION:Transact users with an interactive job submission and execution facility that allows you to do the following:

- Run your VISION:Builder applications.
- Compile/generate your VISION:Transact applications.
- Catalog your definitions in an interactive environment.

This facility automatically generates the necessary JCL for a batch job submission or the necessary CLIST for a foreground job execution. At your request, the generated JCL/CLIST can be automatically submitted, written out to disk for later use, or both.

**Note:** Using the Foreground option, you can route all generated output directly to your terminal for immediate viewing.

The following discussion on the VISION:Workbench Submission facility has been divided into two parts:

- How the Submission facility works.
- Completing the Submission facility data entry panels.

### How the Submission Facility Works

VISION:Workbench uses the ISPF File Tailoring facility to automatically generate your JCL/CLIST. This guide does not attempt to teach you file tailoring, but includes a brief discussion on how file tailoring works to familiarize you with the general concept. For more information on this topic, refer to the ISPF Dialog Management Guide and Reference Guide from IBM®.

The JCL/CLIST generation process starts with a default JCL stream or CLIST called a file tailoring skeleton. [Figure 2-44](#) shows a JCL skeleton example.

```
// &USERID JOB (555,1212,34),'&NAME',  
// MSGCLASS=&MSG,NOTIFY=&USERID  
//*  
//JOB LIB DD DSN=MARKIV. INTEG2.LOADLIB,DISP=SHR  
//*  
//CATALOG EXEC PGM=MARKIV,REGION=1024K  
//M4LIST DD SYSOUT=&LIST  
//M4LIB DD DSN=&LIB,DISP=SHR  
//M4INPUT DD DSN=&INPUT,DISP=SHR  
//*  
//
```

Figure 2-44 JCL Skeleton Example

JCL/CLIST parameters that stay the same from submission to submission are hard-coded in the JCL/CLIST skeleton. Information that is likely to change from submission to submission is represented with ISPF variables. ISPF variables are 1 to 8-character names preceded by an ampersand (&).

Just prior to the actual JCL/CLIST generation process, also known as the file tailoring process, the variables referenced in the skeleton are assigned values. These values are usually obtained by displaying data entry panels that prompt the user for the necessary information.

Once the variable information has been supplied, the file tailoring service can be invoked and the actual JCL/CLIST generation process can take place. During the actual generation process, a copy of the skeleton is made and the variables are replaced with their corresponding values. When this process has finished, the completed JCL/CLIST can be automatically submitted, written to disk for later use, or both.

If you specify that you want to keep the JCL/CLIST, it is automatically written to the data set that is allocated to the ISPF ddname ISPFIL. This data set is often referred to as the “file tailoring output” data set.

- VISION:Builder users must pre-allocate this data set prior to invoking the submission process. If this data set is not pre-allocated, you receive the following message:

THE SPECIFIED SKELETON DOES NOT EXIST.

- VISION:Workbench for ISPF automatically allocates this data set for VISION:Transact users.

To illustrate the concept of JCL/CLIST generation, [Figure 2-45](#) shows a very simple sample skeleton along with its accompanying data entry panel. Notice how the information specified on the data entry panel corresponds to the variables in the skeleton. The completed JCL stream, also shown in the figure, is the end result. Notice that the original skeleton is left unchanged.

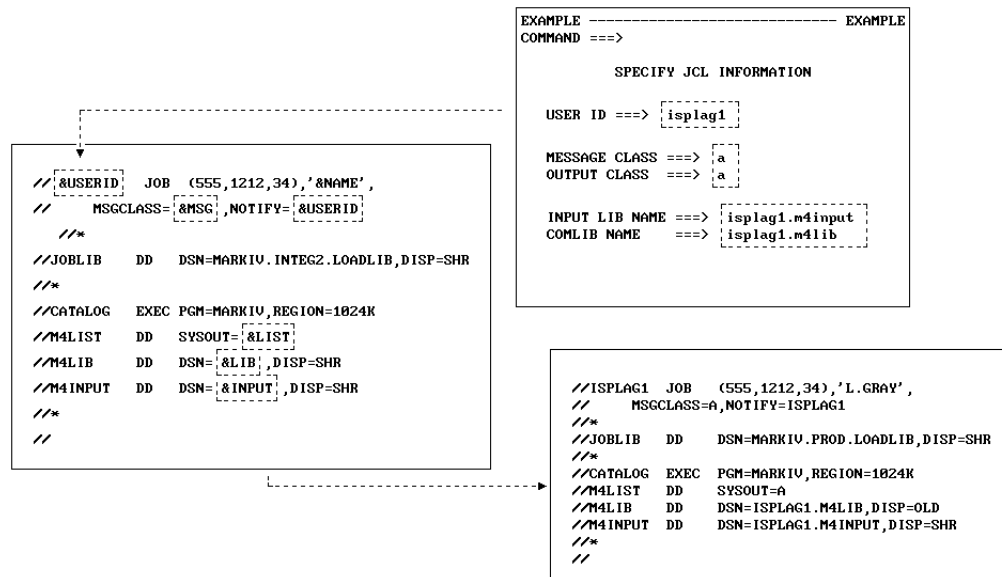


Figure 2-45 JCL/CLIST Generation

Although the VISION:Workbench for ISPF submission skeletons can be more involved than this simple example, they work on the same principle. The variables in the VISION:Workbench for ISPF skeletons are filled in with the information you enter on the submission data entry panels, producing a completed JCL stream or CLIST that can be automatically submitted, saved, or both.

## Using the Submission Facility

Because the submission process is very similar for both VISION:Builder and VISION:Transact, the process for both products is summarized in one discussion.

The Submission facility is entered from the VISION:Workbench Main menu. Selecting a submission option (Foreground or Background options for VISION:Builder or Generate option for VISION:Transact) displays an (Batch or Foreground) Item Entry panel.

**Note:** Foreground or background processing is specified in the Item Entry panel for VISION:Transact.



The figures used in this discussion are from the VISION:Builder Background option (which displays a Batch Item Entry panel). They assume the use of the default VISION:Workbench skeletons and user panels as shipped on your VISION:Workbench for ISPF installation tape. These panels are very similar to the panels used by the other submission options as well. If you understand how these panels work, you will be right at home when using any of the VISION:Builder or VISION:Transact submission facilities.

Once you select a submission (Background, Foreground, or Generate) option, the submission process is as follows:

1. Complete the Item Entry panel. Specify the location of the source object that is to be submitted.
2. Complete the Data Set Specifications panel. This panel supplies information about the files that are needed to actually run the job.
3. Complete the Options panel. This panel supplies processing options for the actual file tailoring process.
4. Complete the User panel. This panel allows you to enter site dependent information that may be needed to successfully complete your JCL and/or CLIST.
5. VISION:Workbench for ISPF generates the necessary JCL or CLIST by invoking ISPF File Tailoring facility.

## Step 1.: Specify Item to Submit

In Step 1. of the submission process, you specify the source member that you want to submit by completing the Item Entry panel shown in [Figure 2-46](#).

```

BATCHITEM ----- BATCH - ITEM ENTRY PANEL -----
COMMAND ===>

ISPF LIBRARY:
  PROJECT ===>
  GROUP   ===>      ===>      ===>      ===>
  TYPE    ===>
  MEMBER  ===>      (Blank for member selection list)

OTHER PARTITIONED OR SEQUENTIAL DATASET:
  DATASET NAME ===>
  VOLUME SERIAL ===>      (If not cataloged)

DATASET PASSWORD ===>      (If password protected)

```

Figure 2-46 The Item Entry Panel

Note that this panel closely resembles a standard ISPF Item Entry panel. Just like an ISPF Item Entry panel, specified data set names that are not enclosed in quotation marks are prefixed with your user ID. If a partitioned data set is entered and a member name is not provided, a member selection list is displayed. Unlike the Edit subsystem, if you are submitting a member from a sequential data set, you do not need to supply an item name.

For VISION:Transact users, this panel is also used to specify either a foreground or background processing mode.

## Step 2.: Supplying File Information

After you specify the source member that you want to submit, VISION:Workbench for ISPF examines the member to determine the type of item being submitted. Based on the item type, and other source information, VISION:Workbench determines the type of run required.

Once a run type is determined, VISION:Workbench determines the file allocate statements (DD statements) that are required by the run. From the information gathered, VISION:Workbench displays a default Data Set Specifications panel. This panel eventually contains the information, or variables, that are used to generate your actual JCL DD statements or CLIST allocation statements.

The default (Batch) Data Set Specifications panel is similar to the panel shown in [Figure 2-47](#).

```

BATCHGEN--- ISPLCB1.MISPF.SOURCE(EMPREF) ----- ROW 1 TO 5 OF 5
COMMAND ==>                                     SCROLL ==> CSR

                                BATCH - DATA SET SPECIFICATIONS

Enter END command to specify job options.
Enter CANCEL command to exit without processing job.

M4DDNAM M4FILENM UNIT DATASET NAME
M4DDOVR  DISPOSITION  VOLSER

M4LIST  _____ SYSOUT A

M4OLD   EMPFILE  SYSDA  _____
        _____ OLD
M4SORT  _____ SYSDA  _____
        _____ NEW,PASS
M4REPO  _____ SYSDA  _____
        _____ NEW,PASS
M4LIB   _____ SYSDA  _____
        _____ SHR

***** BOTTOM OF DATA *****

```

Figure 2-47 A Default (Batch) Data Set Specifications Panel

The actual generated file entries vary depending on the type of run being generated. If there are more entries than can be displayed at one time, you can scroll this panel up and down.

Once you complete this panel, you are ready to proceed to the next step. [Figure 2-48](#) shows what a completed (Foreground) Data Set Specifications panel might look like.

```

FOREGEN--- ISPLCB1.MISPF.SOURCE(EMPREF) ----- ROW 1 TO 5 OF 5
COMMAND ===>                                SCROLL ===> DATA

                                FOREGROUND - DATA SET SPECIFICATIONS

Enter END command to specify job options.
Enter CANCEL command to exit without processing job.

M4DDNAM  M4FILENM  Unit  Dataset name
M4DDOVR          Disposition      Uolser

M4LIST          SYSOUT X_____

M4OLD    EMPFILE  SYSDA  MARKISPF.EMPFILE.DATA_____
          OLD

M4SORT          SYSDA  SORTFILE_____
          NEW,DELETE

M4REPO          SYSDA  RPTFILE_____
          NEW,DELETE

M4LIB          SYSDA  MARKISPF.EMPFILE.M4LIB_____
          SHR

***** BOTTOM OF DATA *****

```

Figure 2-48 A Completed (Foreground) Data Specifications Panel

Notice that SYSOUT data sets are specified by entering “SYSOUT” in the Unit field and specifying the SYSOUT class in the Dataset name field. For foreground jobs, you can route output to your terminal by specifying a SYSOUT data set with a SYSOUT class of (\*).

As you probably noticed, although this is a scrollable display, it does not have line command fields. As a result, you cannot add or delete file entries from the Data Set Specifications panel. If you find that you need to add or delete a file entry, generate the JCL with a processing option of Keep. You can manually edit the generated JCL to complete any final customization before actually submitting the job yourself.

When completing the Foreground Data Set Specifications panel, keep in mind the following:

- All data set names that ARE NOT enclosed in quotation marks are prefixed with your user ID during the execution process.
- Data set names that ARE enclosed in quotation marks are used exactly as specified during the execution process.

When completing the Batch Data Set Specifications panel, remember the following:

- Do not use quotation marks.
- Enter the complete data set name.
- Your user ID is not automatically prefixed to the specified name.

### Step 3.: Completing the Options Panel

After completing the Data Set Specifications panel, you proceed to the Options panel shown in [Figure 2-49](#).

```

BATCHOPT--- ISPDLD1.TCOMM.DEFS(ITEMAPP1) -----
OPTION ==>

                                BATCH - OPTIONS

        K - keep generated job - do not submit
        S - submit job (and delete)
        KS - keep and submit job

Enter option to process job.
Enter END command to exit without submitting job.

YOUR PANEL NAME   ==> M9BGUPNL           Enter the name of a panel to be
                                           displayed prior to file tailoring.

JCL SKELETON NAME ==> M9BGTS             Enter the name of a file tailoring
                                           skeleton JCL member.

FILE TAILORING OUTPUT INFORMATION:      (Options K and KS only)
  OUTPUT FILE MEMBER   ==> EMPREPT
  REPLACE LIKE-NAMED MEMBER ==> YES      (YES or NO)

```

Figure 2-49 The Options Panel

By making the appropriate entry on the command line, you can specify whether or not you want to automatically submit the generated JCL/CLIST (S), keep the generated JCL/CLIST (K), or both (KS).

If you choose to keep the generated JCL/CLIST, the entries on the bottom of the panel allow you to specify member information if you are writing to a partitioned data set.

The information in the middle of the Options panel is where you specify the name of the file tailoring skeleton to be used to generate your JCL/CLIST.

The following table lists the default skeletons and the accompanying user panels that should be used to successfully generate JCL/CLIST using the VISION:Workbench Submission facility as shipped on your installation tape.

	VISION:Builder		VISION:Transact	
	User Panel	Skeleton	User Panel	Skeleton
BACKGROUND	M9BGUPNL	M9BGTS	M9GCTPU1	M9GCTSFG
BACKGROUND	M9BGUPNL	M9BGTS	M9GCTPU2	M9GCTSFG

The Installation and Support Guide contains complete listings of these skeletons, along with listings of the accompanying default user panels.

Along with a skeleton name, the middle section of the Options panel also allows you to specify whether or not you want to display a User panel before the actual JCL/CLIST generation process begins. If you do not need a User panel, leave this entry blank.

## Step 4.: Completing the User Panel

There are many advantages in using the User panel. You can add site dependent variables to your default VISION:Workbench for ISPF submission skeletons. Values can be assigned to these variables during the submission process using the display of a customized User panel.

If you do not need the extra flexibility provided by the User panel, you can replace the default User panel variables found in your default VISION:Workbench for ISPF skeletons with hard-coded constant values. This completely eliminates the need for a User panel display.

[Figure 2-50](#) shows the VISION:Builder batch submission User panel that is supplied with VISION:Workbench for ISPF.

```

BATUPANL - KREJ004.ISPF.TEST(DUPTST) -----
COMMAND ==>

Enter END to process using the option selected on the BATCHOPT panel.
Enter CANCEL command to terminate processing this member.

Enter the name of the VISION:Builder and COMLIB program LOAD LIBRARIES
Builder LOADLIB ==>
COMLIB LOADLIB ==>

Enter the VISION:Builder region size (example 1024K)
Builder REGION ==>

Enter the name of the SORT program LOAD LIBRARY
SORT LOADLIB ==>

SORT SPACE ==>          Number of SORTWORK CYLINDERS
SORT UNIT ==>          SORTWORK UNIT type (example SYSDA)

JOB statement information:
==>
==>
==>

```

Figure 2-50 The VISION:Builder Batch Submission User Panel

To successfully generate a JCL/CLIST using the VISION:Workbench for ISPF Submission facility, as shipped on your installation tape, you must complete the appropriate User panel for batch or foreground submission.

While all of the other submission panels are required, the User panel is optional. Your VISION:Workbench for ISPF Submission facility can be tailored to eliminate the need for the panel. Its purpose is to allow you to specify additional site dependent information that is needed to correctly generate the necessary JCL/CLIST.

## Step 5.: Generating JCL or CLISTs

Once all submission panels have been completed, VISION:Workbench invokes the ISPF File Tailoring services to generate the JCL/CLIST.

## Using the Utilities Subsystem

The VISION:Workbench for ISPF Utilities subsystem provide you with a means for administering and maintaining your COMLIBs in an interactive environment.

Selecting Utilities (option 50) from the VISION:Workbench Main menu displays the Utility Selection Menu panel shown in [Figure 2-51](#).

```
UTILSEL----- UTILITY SELECTION PANEL -----
OPTION ==>

      COMLIB MAINTENANCE FUNCTIONS:

          1  CONDENSE   - Condense A COMLIB
          2  INITIALIZE - Allocate/Initialize A COMLIB
          3  BACKUP    - Backup A COMLIB
          4  RESTORE   - Restore A COMLIB

      ITEM MAINTENANCE FUNCTIONS:

          5  DELETE    - Delete COMLIB Items
          6  COPY      - Copy Items From One COMLIB To Another
          7  DOCUMENT  - Document COMLIB Items
          8  SOURCE    - Retrieve COMLIB Items In Source Form
```

Figure 2-51 The Utility Selection Menu Panel

Note that the utility functions are divided into two groups: COMLIB functions and item functions.

- The COMLIB maintenance functions allow you to perform administration and maintenance activities for an entire COMLIB.
- Item maintenance functions allow you to perform administration and maintenance activities for individual items stored in a COMLIB.

Because the Utilities subsystem is simple and straightforward to use, it is not discussed in detail. Below are a few rules for performing utility functions:

- If you are ever unsure about how to complete a panel, use the interactive Help. The Help facility is invoked using the HELP primary command, usually PF1.
- Whenever you are specifying a data set name, your TSO user ID is prefixed to the data set name if the data set name is not enclosed in quotation marks. If the data set name is enclosed in quotation marks, it is used exactly as specified.
- Any printed output that you specifically request while using the Utilities subsystem is written out to the VISION:Workbench list data set. The VISION:Workbench list data set works exactly as its ISPF list data set counterpart. The VISION:Workbench list data set is discussed in greater detail later in this chapter.
- The ISPF Split Screen primary command is not active in the VISION:Workbench Utilities subsystem. This is done to avoid possible allocation conflicts between the multiple, logical sessions that are created when using Split Screen.

## VISION:Workbench for ISPF Parameters Subsystem

The VISION:Workbench for ISPF Parameters subsystem provides you with information regarding your Computer Associates products.

Selecting Parameters (option 10) from the VISION:Workbench Main menu displays the Parameters Selection Menu panel. The menu contains three option groups and varies depending on which product you are using (VISION:Workbench with VISION:Builder or VISION:Workbench with VISION:Transact).

### *Session Parameters Group*

All menus offer the Session option and the List option in the Session Parameters group. These options allow you to customize certain operating characteristics of your VISION:Workbench for ISPF session.

### *Maintenance Level Group*

The Maintenance Level group Modifications option, also contained on all menus, provides you with a list of all system modifications (SMs) that are currently installed on your VISION:Workbench for ISPF system.

*Default Site Parameters group:*

The Default Site Parameters group allows you to browse the default parameter modules associated with VISION:Builder, VISION:Transact, and COMLIB. These options vary depending on the version of VISION:Workbench for ISPF that you are using.

- When you are using the VISION:Builder version of VISION:Workbench for ISPF, you have options for browsing both the COMLIB parameter module (MARKLIBP) and the VISION:Builder parameter module (M4PARAMS).
- When you are using the VISION:Transact version of VISION:Workbench for ISPF, you have options for browsing the COMLIB parameter module (MARKLIBP), the VISION:Transact parameter module (FIVEPARM), and the VISION:Transact terminal device table.

## User Profile - First Time User Setup

Just as you have a personal user profile for your regular ISPF session, you also have a personal user profile for VISION:Workbench. This profile is automatically created and added to your ISPF profile library (member IADSPROF) the first time that you use VISION:Workbench.

Every ISPF user has a personal profile library associated with their user ID. This library can be thought of as a save area in that it provides a place for ISPF and ISPF applications such as VISION:Workbench to save information about you between sessions. Without a profile library, you would have to reenter all the standard information that you routinely use during your session at the start of each new session. Using a profile library allows this standard session information to be saved at the end of your session. At the start of your next session, the information is automatically retrieved from your profile and placed into the appropriate panels as default entries.

The information that is stored in this user profile is referred to as “session profile information.” For more detailed information on how user profiles work, see IBM's ISPF User's Guide.

Because your VISION:Workbench user profile is empty the very first time that you use VISION:Workbench, take a few minutes to review and complete your user profile information. Once this is done, you do not have to do it again, unless you want to change this information.



VISION:Workbench uses your VISION:Workbench user profile to save the following information about your session:

- VISION:Workbench for ISPF operating characteristics.
- Item entry specifications.
- Validation library specification.
- Screen developer profile.
- Job submission information.

Each of the above areas is discussed in the following subsections.

## Setting Session Parameters

The VISION:Workbench for ISPF Parameters subsystem allows you to customize how certain areas of VISION:Workbench for ISPF work. The Parameters subsystem is invoked by selecting Parameters (option 10) from your product Main menu (PRIMOPT4 or PRIMOPT5). This selection displays the Parameters Selection Menu panel as shown in [Figure 2-52](#).

```

PARMSEL----- PARAMETERS SELECTION PANEL -----
OPTION ==>

SESSION PARAMETERS:

1  SESSION      - Specify Session Parameters
2  LIST         - Specify List File Parameters

MAINTENANCE LEVEL:

3  MODIFICATIONS - Display Applied System Modifications

DEFAULT SITE PARAMETERS:

4  COMLIB       - Display Default COMLIB Parameters
5  VISION:Builder - Display Default VISION:Builder Parameters

```

Figure 2-52 Selecting Session Parameters VISION:Workbench for ISPF with VISION:Builder

To establish user profile information, the Session Parameters group is used. The options in this group directly affect how your VISION:Workbench for ISPF session operates. The options include the Session option (option 1) and the List option (option 2).

Selecting the Session option (option 1) from the Parameters Selection Menu panel displays the Session Parameters panel as shown in [Figure 2-53](#).

PARMSESS-----		SESSION PARAMETERS -----	
COMMAND ==>			
AUTOSKIP ATTRIBUTE FOR FIELDS	==>	OFF	(ON or OFF)
INPUT FIELD PAD CHARACTER	==>		
LINE COMMAND FIELD PAD CHARACTER	==>	'	
DELETE CHARACTER FOR UTILITIES	==>	D	
SEGMENT REFERENCED BY	==>	NAME	(NAME or NUMBER)
FILE/ARRAY LOCATION CALCULATION?	==>	YES	(YES or NO)
TRANSACTION LOCATION CALCULATION?	==>	YES	(YES or NO)
LIBRARY COMPONENT WORKSIZE	==>	256	(Number in K, Minimum 96)
PARAMETER MODULE TO BE USED	==>	M4	(M4=use VISION:Builder parms) (M5=use VISION:Transact parms)
DB2 SUB-SYSTEM ID	==>		
DB2 PLAN NAME	==>		

Figure 2-53 Selecting Session Parameters

The entries on the Session Parameters panel allow you to customize certain operating characteristics of your VISION:Workbench for ISPF session according to your personal preferences.

The AUTOSKIP ATTRIBUTE FOR FILES entry controls cursor positioning while entering data.

- If you specify ON for this entry, your cursor automatically positions to the first byte of the next data entry field, after you complete the last byte of the field you are currently entering. This prevents your keyboard from locking if you accidentally try to type past the end of a field.
- If you specify OFF for this entry, you must use your tab key to move to the next field after having reached the end of the current data entry field. If you accidentally try to type past the end of a field, your keyboard locks.

The INPUT FIELD PAD CHARACTER entry allows you to specify a special character to be used as a pad character for data entry fields that have a value of blank or null. An underscore ( \_ ) is the default pad character for data entry fields.

The LINE COMMAND FIELD PAD CHARACTER entry allows you to specify a special character to be used as a pad character for the line command fields. Line command fields are used to enter VISION:Workbench for ISPF line commands. An apostrophe ( ' ) is the default pad character for line command fields.

All of the VISION:Workbench for ISPF data entry panels in this document use the default pad character of underscore for data entry fields and the default pad character of apostrophe for line command fields.

The DELETE CHARACTER FOR UTILITIES entry allows you to specify a special character that must be used to select COMLIB items for deletion within the VISION:Workbench for ISPF Utilities subsystem. Requiring a special character helps prevent accidental deletions.

The SEGMENT REFERENCED BY entry allows you the choice of defining and referencing file segments by either a Segment Name or a Segment Number.

- If you specify NAME in the SEGMENT REFERENCED BY entry, one variation of the File Definition panel (shown in [Figure 2-54](#)) is displayed when you create a file definition. Notice that this panel only provides for a Segment Name entry. On all subsequent screens that refer back to a segment, you must refer to the segment by its assigned name.

```

FILNALSU ---- MARKISPF.ISPLAG1.SOURCE(EMPFILE) ----- ROW 1 OF 5
COMMAND ==>                                         SCROLL ==> CSR
                                                    FILE: EMPFILE

                        USAM KSDS FILE DEFINITION

      BUFFER SIZE ==> _____ FILE ID           ==> _____
      GLOSSARY    ==> -          EXPIRATION DATE ==> __/__/__
                        USER DATA           ==> _____

      LINE  SEGMENT  HIERARCHICAL  FIXED  SEGMENT
      CMD   NAME      LEVEL         OCCURRENCES  ORDER
      ' ' ' ' ' ' ' ' ' ' ' ' ' ' ' ' ' ' ' ' ' ' ' '
      ' ' ' ' ' ' ' ' ' ' ' ' ' ' ' ' ' ' ' ' ' ' ' '
      ' ' ' ' ' ' ' ' ' ' ' ' ' ' ' ' ' ' ' ' ' ' ' '
      ' ' ' ' ' ' ' ' ' ' ' ' ' ' ' ' ' ' ' ' ' ' ' '
      ' ' ' ' ' ' ' ' ' ' ' ' ' ' ' ' ' ' ' ' ' ' ' '
      ' ' ' ' ' ' ' ' ' ' ' ' ' ' ' ' ' ' ' ' ' ' ' '

      ***** BOTTOM OF DATA *****
  
```

Figure 2-54 File Definitions by Segment Name

- If you specify NUMBER in the SEGMENT REFERENCED BY entry, another variation of the File Definition panel (shown in [Figure 2-55](#)) is displayed when you create a file definition. Notice that this panel has both a Segment Number entry and a Segment Name entry. On all subsequent panels that refer back to a segment, you must identify the segment by its assigned segment number.

- The calculated field start location is based on the start entry and length entry of the previously defined field. The first field is assumed to start in location 1 of the record. To use this feature, the fields must be contiguous within a definition.

- The **LIBRARY COMPONENT WORKSIZE** entry allows you to specify the size of the work area that is used by the Utilities subsystem. If, while using the **VISION:Workbench for ISPF Utilities**, a message is shown informing you that not enough storage is available, increase the amount specified in this entry.

The PARAMETER MODULE TO BE USED entry indicates whether you are using VISION:Builder or VISION:Transact support. This entry automatically defaults to M4 for VISION:Builder customers and M5 for VISION:Transact customers. You should not change this entry.

The DB2 SUB-SYSTEM ID and DB2 PLAN NAME entries pertain to the Edit subsystem's global validation processing of VISION:Builder applications. If you request a global validation of a VISION:Builder application that contains DB2 statements, VISION:Workbench for ISPF uses this information to establish a DB2 environment if one does not already exist. This ensures that your DB2 statements are properly and accurately validated.

Further explanation of the DB2 entries can be found in [VISION:Builder Validation Considerations on page 2-100](#).

## Setting List Data Set Parameters

The VISION:Workbench list data set performs the same function for VISION:Workbench that your ISPF list data set performs for ISPF. Any output that you request during your VISION:Workbench session is written to this data set.

Selecting the List option (option 2) from the Parameters Selection Menu panel displays the Specify List Data Set Defaults panel shown in [Figure 2-56](#). This Parameters subsystem panel allows you to setup default processing information for your VISION:Workbench for ISPF list data set.

```

PARMLOPT----- SPECIFY LIST DATA SET DEFAULTS -----
COMMAND ==>

Process option  ==>
SYSOUT class   ==>
Lines per page ==>
Primary pages  ==>
Secondary pages ==>

VALID PROCESS OPTIONS:
  PD - Print data set and delete      K - Keep data set (without printing)
                                      D - Delete data set (without printing)

JOB STATEMENT INFORMATION:           (Required for system printer)
==>
==>
==>
==>

```

Figure 2-56 Setting List Data Set Defaults

This panel allows you to specify how many lines per page you want printed and how much space, in terms of pages, you want allocated for this data set. This panel also allows you to set up some default JCL information, such as a job statement or route statements. VISION:Workbench for ISPF uses this JCL information when printing your list data set.

## Setting Item Entry and Validation Library Defaults

The first time you use the VISION:Workbench for ISPF Edit subsystem the Item Entry and validation panels are blank, as shown in [Figure 2-57](#).

```

ITEMENTRY ----- DEFINITION - ITEM ENTRY PANEL -----
COMMAND ==>

ISPF LIBRARY:
  PROJECT ==>
  GROUP   ==>
  TYPE    ==>
  MEMBER  ==>
           (Blank for member selection list)

OTHER PARTITIONED OR SEQUENTIAL DATASET:
  DATASET NAME ==>
  VOLUME SERIAL ==>
               (If not cataloged)

COMLIBS:
  COMLIB   ==>
  COMLIB1  ==>
  COMLIB2  ==>
  COMLIB3  ==>
  ITEM NAME ==>
               (Blank for member selection list)

DATASET PASSWORD ==>
               (If password protected)

EXCLUSIVE USE OF PRIMARY COMLIB ==> N (Y=Yes, N=No)

```

```

VALLIBS ----- SPECIFY VALIDATION LIBRARIES -----
COMMAND ==>

Specify the Comlibs that should be used to validate the current item:

  COMLIB   ==>
  COMLIB1  ==>
  COMLIB2  ==>
  COMLIB3  ==>

Indicate whether referenced field names should be cross-checked with
file definitions. This option can be changed during the session
with the NAMES ON and NAMES OFF primary commands.

  NAMES VALIDATION ON? ==>      blank or N = NO
                              Y = YES

Enter END   to Continue processing
Enter CANCEL to Terminate processing of the current item

```

Figure 2-57 First Time Item Entry and Validation Panel

Any information you enter on these panels becomes a default entry. Default entries automatically reappear whenever these panels are used. If you change an existing entry, the new entry replaces the old and becomes the new default entry.

When you end your session, this default information is saved in your user profile. If you tend to use the same set of data sets from session to session, this saves you from having to reenter the same data set names every time you start a new session.

## Setting the Screen Developer Profile

As part of its VISION:Transact support, VISION:Workbench for ISPF offers complete screen painting capability using the Screen Developer subsystem.

When using the Screen Developer to “paint” a VISION:Transact screen, literal and data field display characteristics, such as color and highlighting, are specified by assigning a display attribute character to each literal and/or data field that requires special display characteristics. These display attribute characters are predefined symbols (A–Z and a–z) that have been assigned a set of display characteristics. Because these display attribute characters are used over and over again, VISION:Workbench for ISPF saves these symbols in your user profile.

When you start a VISION:Workbench for ISPF VISION:Transact session and VISION:Workbench for ISPF does not find existing display attribute character definitions in your user profile, the default set of display attribute characters shown in [Figure 2-58](#) is automatically created and added to your user profile.

SCRNPROF ---- MARKISPF . ISPLAG1 . SOURCE (MENU) -----					
COMMAND ==>			SCROLL ==> CSR		
Field Designator ==>			(Default Field Designator - _ )		
Highlight Designator ==> @			(Default Highlight Designator - @ )		
Device Identifier ==>			(Default Device ID: T3270M2 )		
Line Cnd	Attribute Character	Bright/ Dark	Color	Extended HiLite	Program Symbol
0001	B	H	B	-	-
0002	b	-	B	-	-
0003	D	D	-	-	-
0004	G	H	G	-	-
0005	g	G	-	-	-
0006	H	H	-	-	-
0007	K	H	B	-	-
0008	k	-	B	-	-
0009	N	-	N	-	-
0010	P	H	P	-	-
0011	p	-	P	-	-
0012	R	H	R	-	-
0013	r	-	R	-	-
0014	T	H	T	-	-
0015	t	-	T	-	-

Figure 2-58 Screen Developer Default Attribute Characters

Although this default set of display attribute characters does not cover every combination of display characteristics that you may need, it provides you with a base that you can customize to suit your particular needs.

If you are a first time user, enter the Screen Developer as if you are creating a new screen. This ensures that a default set of display attribute characters is created for you. You can review and customize this set of attributes to suit your needs.

When you have completed your customizations, exit the Screen Developer subsystem. This ensures that your customized display attribute character definitions are replaced back into your user profile. The next time that you use the Screen Developer your display attribute characters are set up and ready to use.

## Setting Default Job Submission Information

This section specifically discusses the information that you need to complete the job submission information in the Options panel and the User panel. For a complete discussion on using the Submission options, see [Introduction to Job Submission on page 2-60](#), Introduction to Job Submission, earlier in this guide.

The VISION:Workbench for ISPF Submission facility is entered through the Main menu.

Once entered, most of your job submission information is saved in your VISION:Workbench for ISPF user profile; you do not have to enter it again unless you want to change a specification.

This discussion focuses on the information that you need to provide the first time through. Find out this information before you actually try and use any of the Submission options. Proper generation of your JCL and/or CLISTs depends on proper and accurate information being entered on the job submission data entry panels.

- If your VISION:Workbench for ISPF job submission information has been tailored to meet the particular requirements of your site, consult with the person who completed the customization to determine what information you need to supply.
- If your VISION:Workbench for ISPF job submission information is not customized and you are using it as supplied, the following figures show the default information that you must provide to successfully complete the Options panel and the User panel.

Product	Mode	Options	User Panel
VISION:Builder	Background (Batch)	<a href="#">Figure 2-59</a>	<a href="#">Figure 2-60</a>
VISION:Builder	Foreground	<a href="#">Figure 2-61</a>	<a href="#">Figure 2-62</a>
VISION:Transact	Background	<a href="#">Figure 2-63</a>	<a href="#">Figure 2-64</a>
VISION:Transact	Foreground	<a href="#">Figure 2-65</a>	<a href="#">Figure 2-66</a>

**Note:** “Options panel” is a generic term. The actual Option panel name varies depending on product and mode.

Information that has been specified within the dotted line boxes should be entered exactly as shown.

The dotted line boxes that are not filled in deal with site dependent information. You should consult your system administrator to determine how to complete these entries.



## VISION:Builder Background/Batch Processing

The following figures show how to complete the Options panel and the User panel for the VISION:Builder Background/Batch option.

```

BATCHOPT--- ISPDLD1.TCOMM.DEFS(ITEMAPP1) -----
OPTION  ===>

                                BATCH - OPTIONS

        K - keep generated job - do not submit
        S - submit job (and delete)
        KS - keep and submit job

Enter option to process job.
Enter END command to exit without submitting job.

YOUR PANEL NAME  ===> M9BGUPNL      Enter the name of a panel to be
                                   displayed prior to file tailoring.

JCL SKELETON NAME ===> M9BGTS      Enter the name of a file tailoring
                                   skeleton JCL member.

FILE TAILORING OUTPUT INFORMATION:      (Options K and KS only)
  OUTPUT FILE MEMBER  ===> EMPREPT
  REPLACE LIKE-NAMED MEMBER ===> YES      (YES or NO)

```

Figure 2-59 VISION:Builder Background Options Panel

```

BATUPANL - KREJ004.ISPF.TEST(DUPTST) -----
COMMAND ===>

Enter END to process using the option selected on the BATCHOPT panel.
Enter CANCEL command to terminate processing this member.

Enter the name of the VISION:Builder and COMLIB program LOAD LIBRARIES
Builder LOADLIB  ===>
COMLIB LOADLIB   ===>

Enter the VISION:Builder region size (example 1024K)
Builder REGION   ===>

Enter the name of the SORT program LOAD LIBRARY
SORT LOADLIB     ===>

SORT SPACE       ===>                Number of SORTWORK CYLINDERS
SORT UNIT        ===>                SORTWORK UNIT type (example SYSDA)

JOB statement information:
===>
===>
===>

```

Figure 2-60 VISION:Builder Background User Panel

## VISION:Builder Foreground Processing

The following figures show how to complete the Options panel and the User panel for the VISION:Builder Foreground option.

```

FOREOPTS --- 'KREJ004.ISPF.TEST'(DUPTST) -----
OPTION  ==> K

                                FOREGROUND - OPTIONS

      K - keep generated job - do not submit
      S - submit job (and delete)
      KS - keep and submit job

Enter one of the above options to process the job.
Enter END command to exit without submitting the job.

YOUR PANEL NAME      ==> M9FCUPNL      Enter the name of a panel to be
                                         displayed prior to file tailoring.

CLIST SKELETON NAME ==> M9FCTS        Enter the name of a file tailoring
                                         skeleton clist.

FILE TAILORING OUTPUT INFORMATION:      (Options K and KS only)
  OUTPUT FILE MEMBER      ==> DUPTST
  REPLACE LIKE-NAMED MEMBER ==> NO      (YES or NO)

```

Figure 2-61 VISION:Builder Foreground Options Panel

```

FORUPANL- 'KREJ004.ISPF.TEST'(DUPTST) -----
COMMAND ==>

Enter END to process using the option selected on the FOREOPTS panel.
Enter CANCEL command to terminate processing this member.

Enter the name of the VISION:Builder LOAD LIBRARY
VISION:Builder LOADLIB ==>

Enter the name of the SORT program LOAD LIBRARY
SORT  LOADLIB ==>

SORT SPACE      ==>      Number of SORTWORK CYLINDERS
SORT UNIT       ==>      SORTWORK UNIT type ( example SYSDA )

```

Figure 2-62 VISION:Builder Foreground User Panel

## VISION:Transact Background Processing

The following figures show how to complete the Options panel and the User panel for the VISION:Transact Background option.

```

BGOPT--- KREJ004.ISPF.TEST(DUPTST)-----
OPTION ==> K

          K - KEEP GENERATED JCL - DO NOT SUBMIT
          S - SUBMIT THE JCL AND THEN DELETE IT
          KS - SUBMIT THE JCL AND KEEP IT

ENTER ONE OF THE ABOVE OPTIONS TO PROCESS THE TASK.
ENTER THE END COMMAND TO EXIT WITHOUT FURTHER PROCESSING.

SKELETON NAME      ==> M9GCTSBC      ENTER THE NAME OF THE ISPF FILE
                                     TAILORING SKELETON TO BE USED.

YOUR PANEL NAME    ==> M9GCTPU2     ENTER THE NAME OF A PANEL TO BE
                                     DISPLAYED PRIOR TO FILE TAILORING.

FILE TAILORING OUTPUT INFORMATION:
  DATA SET NAME ==> 'KREJ004.ISPF.TEST(DUPTST)'
  REPLACE LIKE-NAMED MEMBER ==> YES

JOB STATEMENT INFORMATION:
==>
==>
==>
==>

```

Figure 2-63 VISION:Transact Background Options Panel

```

USRPNEL --- KREJ004.ISPF.TEST(DUPTST) -----
COMMAND ==>

ENTER END TO PROCESS USING THE OPTION PREVIOUSLY ENTERED.
ENTER CANCEL TO TERMINATE PROCESSING THIS MEMBER.

ENTER THE NAME OF THE VISION:TRANSACT GEN LIBRARY:
VISION:TRANSACT LOADLIB ==>

ENTER THE NAME OF THE COMLIB LOAD LIBRARY:
COMLIB LOADLIB ==>

LINK OBJECT FILE?      ==>          (YES OR NO)
  LINK TO              ==>
  INCLUDES FROM:
    VISION:TRANSACT    ==>
    MONITOR            ==>

```

Figure 2-64 VISION:Transact Background User Panel

## VISION:Transact Foreground Processing

The following figures show how to complete the Options panel and the User panel for the VISION:Transact Foreground option.

```

FOROPT--- KREJ004.ISPF.TEST(DUPTST)-----
OPTION ==> K

        K - KEEP GENERATED CLIST - DO NOT EXECUTE
        S - EXECUTE THE CLIST AND THEN DELETE IT
        KS - EXECUTE THE CLIST AND KEEP IT

ENTER ONE OF THE ABOVE OPTIONS TO PROCESS THE TASK.
ENTER THE END COMMAND TO EXIT WITHOUT FURTHER PROCESSING.

SKELETON NAME      ==> M9GCTSGC      ENTER THE NAME OF THE ISPF FILE
                                     TAILORING SKELETON TO BE USED.

YOUR PANEL NAME    ==> M9GCTPUL      ENTER THE NAME OF A PANEL TO BE
                                     DISPLAYED PRIOR TO FILE TAILORING.

FILE TAILORING OUTPUT INFORMATION:
DATA SET NAME ==> 'KREJ004.ISPF.TEST(DUPTST)'
REPLACE LIKE-NAMED MEMBER ==> YES

```

Figure 2-65 VISION:Transact Foreground Options Panel

```

USRPNEL --- KREJ004.ISPF.TEST(DUPTST) -----
COMMAND ==>

ENTER END TO PROCESS USING THE OPTION PREVIOUSLY ENTERED.
ENTER CANCEL TO TERMINATE PROCESSING THIS MEMBER.

ENTER THE NAME OF THE VISION:TRANSACT GEN LIBRARY:
VISION:TRANSACT LOADLIB ==>

LINK OBJECT FILE?      ==>          (YES OR NO)
LINK TO                ==>
INCLUDES FROM:
    VISION:TRANSACT    ==>
    MONITOR            ==>

```

Figure 2-66 VISION:Transact Foreground User Panel

## Allocating ISPFIL

If you specify the K (for Keep) or KS (for Keep/Submit) options on the Options panel during the VISION:Workbench for ISPF job submission process, the generated JCL or CLIST is automatically written to the data set that is allocated to the ISPF ddname ISPFIL. This data set is often referred to as the file tailoring output data set.

## Allocating ISPFIL for VISION:Transact

VISION:Transact users do not need to allocate a file tailoring output data set. VISION:Workbench for ISPF automatically allocates the output data set specified on the Options panel.

You should keep the following things in mind when completing this panel:

- If you DO NOT enclose this data set name in quotation marks, your user ID is prefixed to the specified name.
- If you DO enclose this data set name in quotation marks, it is used exactly as entered on the panel.

## Allocating ISPFIL for VISION:Builder

VISION:Builder users (who plan to use the submission Keep or Keep/Submit processing options) must pre-allocate their file tailoring output data set. It must be allocated prior to using either of the VISION:Builder submission options. The VISION:Builder submission options DO NOT automatically allocate this data set for you.

If you do not pre-allocate this file, you receive the following error message on the Options panel:

THE SPECIFIED SKELETON DOES NOT EXIST.

You can automatically allocate or manually allocate ISPFIL:

- Automate the pre-allocation of a file tailoring output data set by including an ISPFIL allocation in your ISPF startup CLIST. This ensures that your file tailoring output data set is always allocated and ready to use whenever you are using VISION:Workbench for ISPF.
- Manually allocate a file tailoring output data set using the TSO ALLOCATE command. This allocation only needs to be done once per VISION:Workbench for ISPF VISION:Builder session. The example below shows how this can be done from your VISION:Workbench for ISPF Main menu.

```
PRIMOPT4 ----- VISION:Builder DEVELOPMENT FACILITY -----
OPTION ==> TSO ALLOC F(ISPFIL) DA('MARKISPF.ISPLCB1.FTOUTPUT')

SPECIFY/REVIEW VISION:Workbench PARAMETERS:

  10 PARAMETERS - Specify Or Display Parameters

EDIT VISION:Builder APPLICATIONS:          EDIT COMLIB DEFINITIONS / UTILITIES:

  20 APPLICATION - Edit Application          40 FILE          - Edit File Def
  21 PROCEDURE  - Edit Procedure            41 LDU           - Edit LDU Def
  25 REQUEST    - Edit Request              42 ARRAY         - Edit Array Def
                                           43 TABLE        - Edit Table Def
  30 FOREGROUND - Execute Foreground        44 TRANSACTION  - Edit Tran Def
  31 BACKGROUND - Submit Background         50 UTILITIES    - Access COMLIB
                                           51 IMPORT       - Quick Starts

VISION:Workbench FOR ISPF DEVELOPMENT FACILITY:
  T TUTORIAL   - View The VISION:Workbench Tutorial
  X EXIT       - Exit VISION:Workbench
```

Figure 2-67 Using the TSO ALLOCATE Command

**Note:** You must create the file tailoring output data set before you allocate it.

File Tailoring Output Data Set Characteristics

The file tailoring output data set must be a partitioned data set with the following characteristics:

- DSORG: PO
- RECFM: FB
- LRECL: 80
- BLKSIZE: Any multiple of 80 (for example, 3120)

This configuration is acceptable for both JCL and CLIST generation.

For additional information on allocating a file tailoring output data set, see the appropriate product installation guide.

## The List Data Set

The VISION:Workbench list data set corresponds to the ISPF list data set in purpose and function. It is used to hold all output that you request during your VISION:Workbench session. For example, if you use the VISION:Workbench Utilities subsystem to perform a source statement retrieval and you request a hard-copy of the retrieved source, the retrieved source listing is automatically written to this data set.

The VISION:Workbench for ISPF list data set must be allocated to the ddname M9LIST. You can pre-allocate this data set prior to invoking VISION:Workbench for ISPF, but pre-allocation is not mandatory. If VISION:Workbench for ISPF finds that this data set has not been pre-allocated, it dynamically allocates it for you, as needed, using the following naming convention:

`&SYSPREF.(&SYSUID.)M9TEMPn.LIST`

The &SYSUID qualifier is only used if different from the system prefix, SYSPREF.

If you choose to pre-allocate M9LIST, you can allocate it as a SYSOUT data set or you can allocate it with the following characteristics:

- DSORG = PS
- RECFM = FBA
- LRECL = 133
- BLKSIZE = 1330

When you terminate a VISION:Workbench for ISPF session in which the list data set was *pre-allocated* prior to invoking VISION:Workbench for ISPF, VISION:Workbench for ISPF DOES NOT perform any kind of termination processing on this data set. It is up to you to print and/or delete this data set as needed.

On the other hand, if the utility list data set was *dynamically allocated* by VISION:Workbench for ISPF, the Process List Dataset panel, shown in [Figure 2-68](#), is displayed at the end of your VISION:Workbench for ISPF session.

```
LDSDISP ----- PROCESS LIST DATASET -----
COMMAND ==>

Process option    ==> PD
SYSOUT class     ==> A

VALID PROCESS OPTIONS:
    PD - Print dataset and delete          K - Keep data set (without printing)
                                           D - Delete data set (without printing)

INSTRUCTIONS:
    Press ENTER key to complete termination.
    Enter END command to return to the primary option menu.

JOB STATEMENT INFORMATION:                (Required for system printer)
==> //ISPLAG1A JOB (102010,279300,SPG,37),'L.GRAY-BN36',
==> //      MSGCLASS=X,NOTIFY=ISPLAG1
==>
==>
```

Figure 2-68 The Process List Dataset Panel

The Process List Dataset panel functions in the same manner as the ISPF List Dataset panel. It allows you to tell VISION:Workbench for ISPF how to process this data set. You can request that VISION:Workbench for ISPF automatically print this data set for you by specifying a processing option of PD (print and delete).

You can set up default processing parameters for this screen using the VISION:Workbench for ISPF Parameters subsystem.



## Using Navigation Short Cuts

Navigation shortcuts include command stacking, using the jump facility, and making multiple line command selections.

### Using Command Stacking

Once you become familiar with the various VISION:Workbench for ISPF option numbers you can start taking advantage of command stacking. Command stacking is an ISPF feature that allows you to enter multiple primary commands.

For example, if you are in the Main menu and you want to change some of your VISION:Workbench for ISPF session parameters, you would normally go through the series of panel displays shown in [Figure 2-69](#).

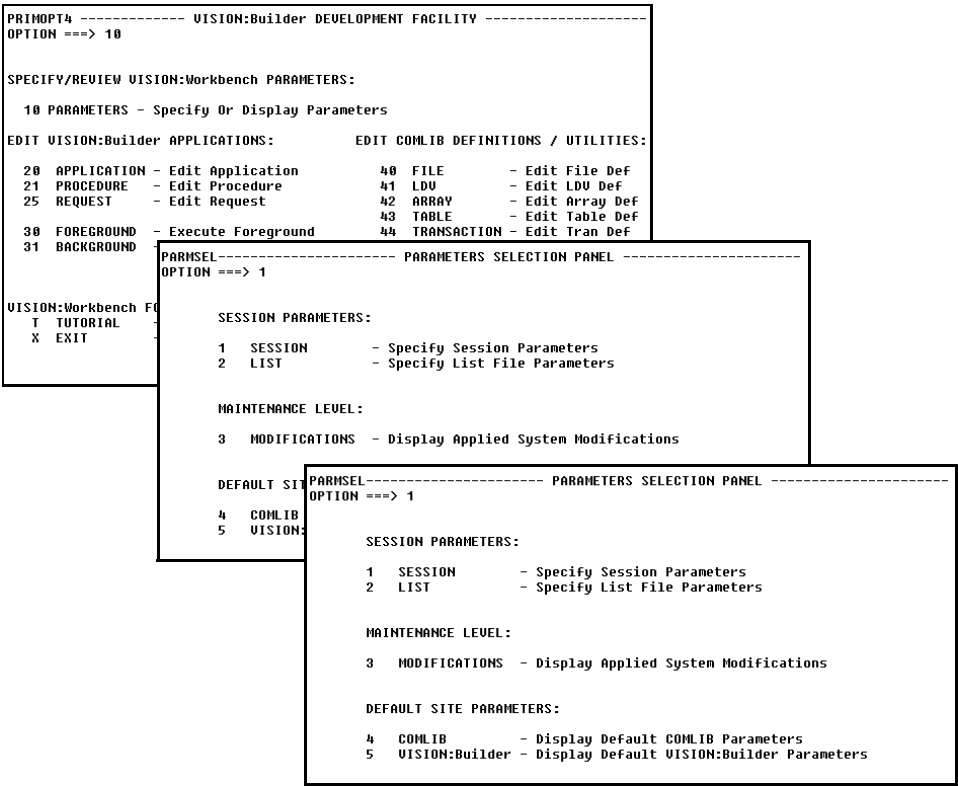


Figure 2-69 Using Sequential VISION:Workbench Panel Displays

If you are already familiar with the session parameter option number, you can shorten this process by taking advantage of command stacking, as shown in [Figure 2-70](#).

```

PRIMOPT4 ----- VISION:Builder DEVELOPMENT FACILITY -----
OPTION ==> 10.1

SPECIFY/REVIEW VISION:Workbench PARAMETERS:

  10 PARAMETERS - Specify Or Display Parameters
EDIT VISION:Builder APPLICATIONS:          EDIT COMLIB DEFINITIONS / UTILITIES:

  20 APPLICATION - Edit Application          40 FILE - Edit File Def
  21 PROC
  25 REQUIRE PARMSSESS----- SESSION PARAMETERS -----
  30 FORE COMMAND ==>
  31 BACK

VISION:Workbench TUTORIAL X EXIT

AUTOSKIP ATTRIBUTE FOR FIELDS ==> ON (ON or OFF)
INPUT FIELD PAD CHARACTER ==> _
LINE COMMAND FIELD PAD CHARACTER ==> .
DELETE CHARACTER FOR UTILITIES ==> D
SEGMENT REFERENCED BY ==> NUMBER (NAME or NUMBER)
FILE/ARRAY LOCATION CALCULATION? ==> NO (YES or NO)
TRANSACTION LOCATION CALCULATION? ==> NO (YES or NO)
LIBRARY COMPONENT WORKSIZE ==> 256 (Number in K, Minimum 96)
PARAMETER MODULE TO BE USED ==> M4 (M4=use VISION:Builder parms)
(M5=use VISION:Transact parms)
DB2 SUB-SYSTEM ID ==>
DB2 PLAN NAME ==>

```

Figure 2-70 Using Command Stacking

Notice how command stacking enables you to bypass the display of the Parameters subsystem main menu and go directly to the Session Parameters panel.

The previous example is a simple one. As you become familiar with the various options within the VISION:Workbench for ISPF Edit subsystem, you can use command stacking to a much greater advantage in moving up and down through the panel hierarchy.

Unlike ISPF, VISION:Workbench allows you to take command stacking one step further in that you can specify an item name at the end of a command stack. This allows you to bypass the Item Entry panel when retrieving an item for edit. When bypassing the Item Entry panel, VISION:Workbench automatically pulls the specified item from your default library. The default library is the last library that you used on the Item Entry panel.

For example, [Figure 2-71](#) shows the series of panels you would normally see when preparing to edit a file definition

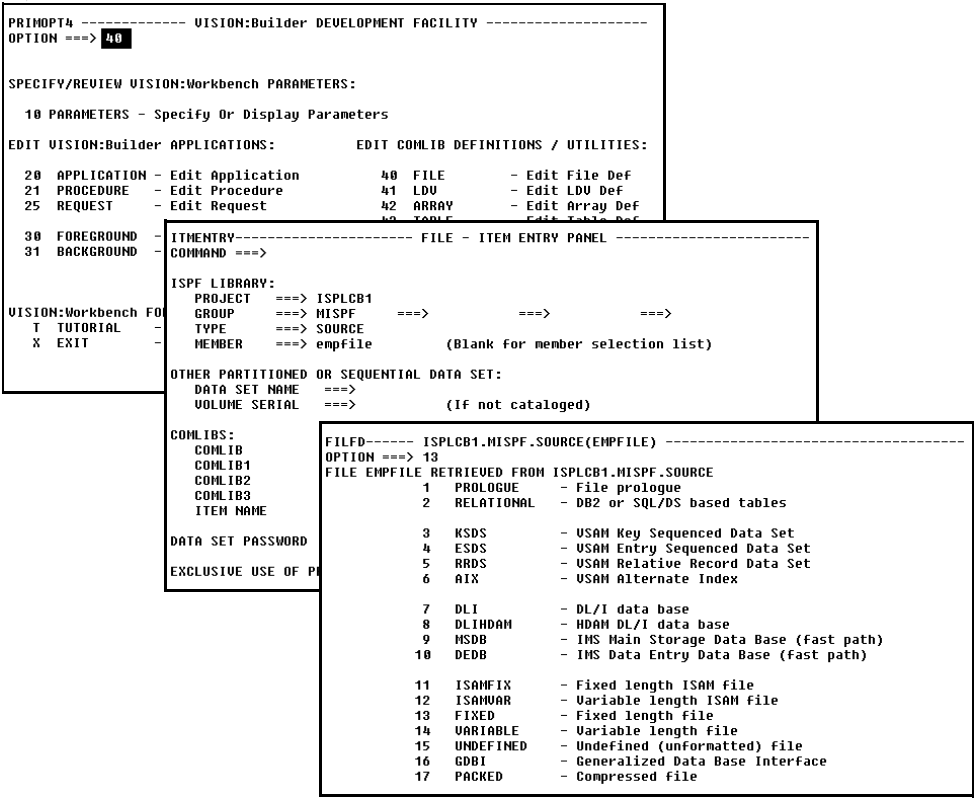


Figure 2-71 Normal File Definition Panel Sequence

Figure 2-72 shows how this process can be shortened by taking advantage of command stacking, which appends an item name specification at the end of the command stack.

```
PRIMOPT4 ----- VISION:Builder DEVELOPMENT FACILITY -----
OPTION ==> 40 empfile

SPECIFY/REVIEW VISION:Workbench PARAMETERS:

  10 PARAMETERS - Specify Or Display Parameters

EDIT VISION:Builder APPLICATIONS:          EDIT COMLIB DEFINITIONS / UTILITIES:

20 APPLICATION - Edit Application           40 FILE          - Edit File Def
21 PROCEDURE  - Edit Procedure             41 LDB           - Edit LDB Def
25 REQ
30 FOR
31 BAC
FILE EMPFILE RETRIEVED FROM ISPLCB1.MISPF.SOURCE
OPTION ==> 13
1 PROLOGUE - File prologue
2 RELATIONAL - DB2 or SQL/DS based tables
3 KSDS      - USAM Key Sequenced Data Set
4 ESDS      - USAM Entry Sequenced Data Set
5 RRDS      - USAM Relative Record Data Set
6 AIX       - USAM Alternate Index
7 DLI       - DL/I data base
8 DLIHDAM   - HDAM DL/I data base
9 MSDB      - IMS Main Storage Data Base (fast path)
10 DEDB     - IMS Data Entry Data Base (fast path)
11 ISAMFIX  - Fixed length ISAM file
12 ISAMVAR  - Variable length ISAM file
13 FIXED    - Fixed length file
14 VARIABLE - Variable length file
15 UNDEFINED - Undefined (unformatted) file
16 GDBI     - Generalized Data Base Interface
17 PACKED   - Compressed file
```

Figure 2-72 Using Command Stacking with a Member Name



Now, from the Application Files panel, assume you want to jump directly to the Request and ASL Procedures panel. [Figure 2-74](#) shows how you can do this using the Jump function.

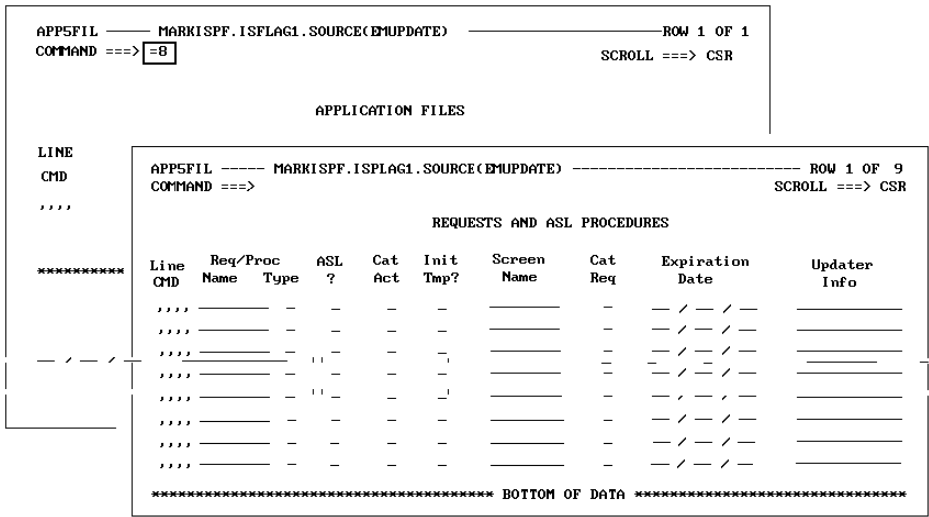


Figure 2-74 Jump Sequence - Application Files to Request and ASL Procedures Panel

The complete series of panels normally required to complete this is shown in [Figure 2-75](#).

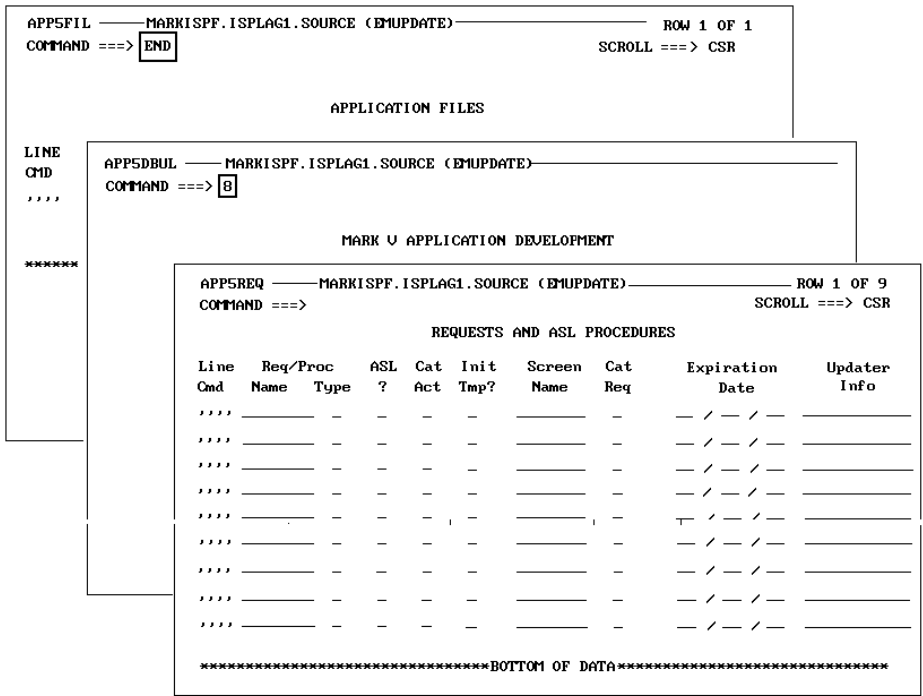


Figure 2-75 Normal Sequence - Application Files to Request and ASL Procedures Panel

The important thing to remember when using the Jump function is that you can only jump between options that are under the same primary menu. Primary menus for VISION:Workbench with VISION:Builder and VISION:Workbench with VISION:Transact include:

- Main menu or Primary Option menu
- Application menu
- Request menu

The following table lists the VISION:Workbench primary menus.

Primary Menu Name	Primary Menu Description
PRIMOPT4	VISION:Builder Main or Primary Option Menu
APP4DEVL	VISION:Builder Application Development Menu
REQ4MENU	VISION:Builder Request Parts Menu
PRIMOPT5	VISION:Transact Main or Primary Option Menu
APP5DEVL	VISION:Transact Application Development Menu
REQ5MENU	VISION:Transact Request Parts Menu
SCRNMENU	VISION:Transact Screen Developer Menu

Note that navigation through the VISION:Workbench for ISPF panel hierarchy becomes faster as you become more familiar with the available options.

## Using Multiple Line Command Selections

VISION:Workbench for ISPF allows you to select multiple rows of data for continued processing at one time. If you select multiple rows, VISION:Workbench for ISPF automatically processes each selected row in sequence, one after another.

For example, to enter additional field information, you can use the Select line command to select multiple field rows. As shown in [Figure 2-76](#), VISION:Workbench for ISPF processes all of the selected field rows before returning to the Field Definitions panel.



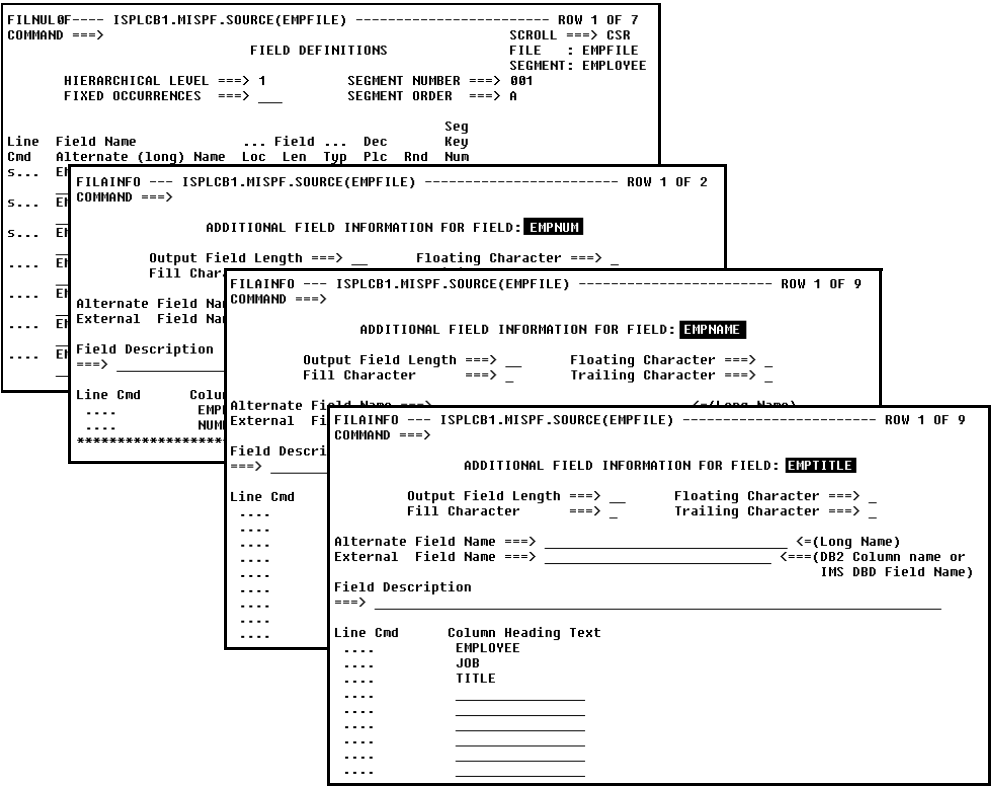


Figure 2-76 Making Multiple Line Command

Multiple Select line commands can be used on any panel that has a scrollable data entry area containing line command fields.

## Maintaining Existing Source

In addition to creating new application systems, VISION:Workbench for ISPF can be used to maintain and enhance your existing VISION:Builder and VISION:Transact application systems.

The process of loading an existing source item into VISION:Workbench for ISPF for modification is called *importing*. The process of writing the modified source back out to disk is called *exporting*.

When importing existing source, the following guidelines apply to all VISION:Builder and VISION:Transact items:

- VISION:Workbench for ISPF correctly imports any item that was generated using VISION:Workbench for ISPF or VISION:Workbench for DOS.
- VISION:Workbench for ISPF also imports existing items that are not generated under VISION:Workbench for ISPF. When importing items that are not generated by VISION:Workbench for ISPF, you should adhere to the following guidelines. If these guidelines are not followed, the import process can produce unpredictable results.

### VISION:Builder and VISION:Transact Source Importing Guidelines

The following describes guidelines for importing source not generated by VISION:Workbench for ISPF:

- When retrieving source from a partitioned data set, the member name must be identical to the item name specified in the first eight positions of the run control information. If these two names are not identical, VISION:Workbench for ISPF automatically changes the run control item name to match the specified member name.
- The item should be syntactically correct.
- The item must only contain one item of the selected type. For example, when importing a file definition, the source member being imported can only contain one file definition. When importing an application, the source member being imported can only contain one application. Applications can, however, contain multiple instream definitions.
- Definition delete statements (“Y” in position 19) are commonly placed in front of a definition to ensure that the existing version of the definition is deleted before the new version is cataloged. However, you should be aware that VISION:Workbench for ISPF ignores these delete statements. They are not imported. As a result, avoid the use of delete statements and use the replace operator instead.

- The source should contain source statements that are placed in the same order as shown on your formatted source statement listings. When formatted source statement listings are produced, statements are sequenced in what is considered to be the most efficient processing order. In general, this is the same statement order that VISION:Workbench for ISPF expects to encounter during the import process. Trying to import existing source members that contain “out of order” statements can produce unpredictable results.
- All statements with the same name in columns 1-8 must be grouped together.
- All IT statements should be placed at the beginning of the item.
- ASL Procedures must have a ##PROC statement immediately following the ER statement (where, # is the system delimiter) and a ##PEND statement at the end of the procedure.

### Importing Source for VISION:Builder Only

The following describes guidelines for importing VISION:Builder source:

- All processing and temporary field specifications must come before the report specifications.
- Each report must start with an En statement that must be followed by the remaining report specification statements for that report.
- Definition replace specification is preserved in the following definition types: stand-alone file definitions, logical data view definitions, array definitions, table definitions, and transaction definitions. The replace specification is *not* preserved in file definitions within an application. Any value other than an “R” is dropped during the import process.

### Importing Source for VISION:Transact Only

All VISION:Transact requests must start with an ER statement.

### VISION:Transact Conversions

When importing existing VISION:Transact applications with a Run Type of A on the RC statement (column 38), VISION:Workbench for ISPF automatically converts the Run Type to G and sets the Library Open entry (28) to U.

### Converting Run Control Information

The most significant change in the run control statement conversion is that, if present, file information is removed from the run control (RC) statement and placed on a run file (RF) statement instead.

- If not already present, OS (output screen) operators are generated for all screen control statements that contain an output screen entry. Note that OS operator stands for output screen operator. (See [Figure 2-77.](#))
- In an effort to encourage the use of screen control statements to tie request processing to a particular screen, all existing screen name entries are removed from request statements. The appropriate screen control statements are generated to tie the request being processed to its corresponding screen.
  - If the request is an input request, a screen control statement with an XR operator is generated.
  - If the request is an output request, the request name is added to the appropriate output screen control statements. (See [Figure 2-77.](#))

[illegible]

Figure 2-77 A Converted Output Request



# VISION:Builder Validation Considerations

The global validation of your VISION:Builder application is actually accomplished by invoking a VISION:Builder scan run. The generated output from this scan run is written to one of the VISION:Workbench for ISPF internal work data sets. When the scan run has completed, you are able to examine the validation results by browsing the work data set, as shown in [Figure 2-80](#).

```
BROWSE      KREJ004.M$TEMP1.LIST                      Line 00000433 Col 001 080
Command ==>                                         Scroll ==> CSR
(R1)          (STATE  ) 001-1
-----
STMT  SEQ
TYPE NO.                                     TEXT
-----
(T1)                                     (TEST OF THE VALIDATION RUN
(T1)                                     (END OF TITLE#
MAR 28, 2002 17.52.04
** MK4YC00 TYPE 0 SORT CONTROL STMT - SORT FIELDS=(5,10,A),FORMAT=BI,FILSZ=16
** MK4YC00 TYPE 0 SORT CONTROL STMT - RECORD TYPE=V,LENGTH=(4092,1290,4092,25
** MK4YC00 TYPE 0 SORT CONTROL STMT - END
** MK4W204 TYPE 0 NUMBER OF MESSAGES PRINTED IS 4.
                        M4OLD - 3 RECORDS INPUT
                        M4REPO - 16 RECORDS OUTPUT
                        M4SORT - 3 RECORDS OUTPUT
                        M4INPUT - 25 RECORDS INPUT
                        M4LIST - 123 RECORDS OUTPUT
                        2 TRACKS ASSIGNED TO M4LIB -- 1 TRACKS NOT FULL. LI
** MK4W209 TYPE 0 1015392 BYTES OF MAIN STORAGE UNUSED DURING DECODING PHASE
***** Bottom of Data *****
```

Figure 2-80 Browsing VISION:Builder Validation Results

When you are done browsing this data set, VISION:Workbench for ISPF automatically deletes the listing.

## DB2 Considerations

If your VISION:Builder application contains any DB2 files, a DB2 environment is required to successfully complete a global validation. If a DB2 environment does not exist prior to invoking the validation process, VISION:Workbench for ISPF uses the DB2 information specified in your session parameters to temporarily establish a DB2 environment.

Use the following entries on the Session Parameters panel to set up default DB2 environment information.

PARMSESS-----

SESSION PARAMETERS -----

COMMAND ==>

AUTOSKIP ATTRIBUTE FOR FIELDS

==> OFF

(ON or OFF)

INPUT FIELD PAD CHARACTER

==>

LINE COMMAND FIELD PAD CHARACTER

==>

`

DELETE CHARACTER FOR UTILITIES

==> D

SEGMENT REFERENCED BY

==> NAME

(NAME or NUMBER)

FILE/ARRAY LOCATION CALCULATION?

==> YES

(YES or NO)

TRANSACTION LOCATION CALCULATION?

==> YES

(YES or NO)

LIBRARY COMPONENT WORKSIZE

==> 256

(Number in K, Minimum 96)

PARAMETER MODULE TO BE USED

==> M4

(M4=use VISION:Builder parms)

(M5=use VISION:Transact parms)

DB2 SUB-SYSTEM ID

==>

DB2 PLAN NAME

==>

Figure 2-81 Specifying DB2 Environment Information

**Note:** If you plan on including DB2 files in your VISION:Builder application, it is extremely important that you complete these DB2 session parameters. If you attempt to validate a VISION:Builder application that contains DB2 files without the proper DB2 environment being set up, your VISION:Workbench for ISPF session will ABEND.

Sample Reports

To create a sample report on the Application Parameter panel (APP4PARM), enter an R (scan only with sample reports) in the Scan/Terminate Control entry. (See [Figure 2-82](#).)

For GDBI, enter N (scan only, mapping decode rules, sample reports) in the Scan/Terminate Control entry. A sample report is included in the global validation output.

```

APP4PARAM---- ISPLCB1.MISPF.SOURCE(NEW) -----
COMMAND ==>

      RUN CONTROL                                RUN PARAMETERS

SINGLE BUFFER ALL FILES ==> -      SYSDATE                                ==> _____
SORT CONTROL           ==> -      SYSDATE4                                ==> _____
OVERRIDE DELIMITER     ==> -      FREESIZE                                ==> _____
SCAN/TERMINATE CONTROL ==> R      AUTOMATIC GRAND SUMMARIES? ==> -
SOURCE STATEMENT LISTING ==> -      MAXIMUM GETMAIN SIZE                ==> _____
SPOOL BLOCKING FACTOR  ==> -      COMPILE REPORT STEP?                  ==> -
                                                                    REPORTER STORAGE SIZE          ==> _____
STORAGE OPTIMIZATION?  ==> -      SORT STORAGE SIZE                      ==> _____
MESSAGE OPTIMIZATION?  ==> -      BYPASS FOLLOW-UP CORD PASS? ==> -
REPORT FILE OPTIMIZATION ==> -      LIST CONTROL OPTIONS?              ==> -
                                                                    ONE BUFFER FOR MASTER FILE?   ==> -
                                                                    ABEND                          ==> _____
                                                                    DUMP?                          ==> -
      OWN CODE                                INHIBIT MULTIPLE LIBRARIES? ==> -
DYNAMIC OWN CODE HOOKS? ==> -      DB2 SQL ID                          ==> _____
                                                                    DB2 SUBSYSTEM ID              ==> _____
      DISPLAY INFO MESSAGES                DB2 PLAN NAME                  ==> _____
DECODE PHASE MSGS      ==> -      DB2 EXPLAIN QUERY NUMBER          ==> _____
PROCESSING PHASE MSGS  ==> -
REPORT PHASE MSGS      ==> -
  
```

Figure 2-82 Requesting a Sample Report in the Application Parameter Panel

In order to process application validation requests that involve sample reports, it is necessary to make some temporary modifications to the application during the validation. These alterations are necessary to avoid the overhead that would be involved in dynamically allocating alternate list and alternate report files. Although these changes do not affect the actual application, they are included in the validation listing that you browse after the validation run has completed.

When processing an application validation run that includes sample reports, the following alterations are made to a copy of your application:

- If the Scan/Terminate Control entry is R (scan only with sample reports) and you have specified the use of an alternate report file in your report parameter specifications, VISION:Workbench for ISPF temporarily sets the Source Statement Listing entry to D (report output to M4LIST1).

The Scan/Terminate and Source Statement Listing entries are found on the Application Parameter panel, APP4PARAM.

- If the Scan/Terminate Control entry is N (scan only, mapping decode rules, sample reports), VISION:Workbench for ISPF temporarily sets the Source Statement Listing entry to E (report output to include both application and mapping request statements).

The Scan/Terminate and Source Statement Listing entries are found on the Application Parameter panel, APP4PARAM.



- Application file entries that have a File Usage entry of M4REPn, are not included in the application validation run.

This File Usage entry is found on the Application Files panel, APP4FIL.

- All Report Handling entries of RF (alternate report file) are temporarily replaced by a Report Handling entry of AL (alternate list file).

The Report Handling entry is found on the Report Output Specifications panel (REQ4REPT), within the Application Request option of VISION:Workbench for ISPF.

## VISION:Transact Validation Considerations

The global validation of your VISION:Transact application is accomplished by invoking an actual VISION:Transact application generation (APPGEN) run. The generated output from this APPGEN run is written to a VISION:Workbench for ISPF internal work data set. If any errors are encountered during the validation, you can examine the validation results by browsing the work data set as shown in [Figure 2-83](#).

BROWSE -- ISPLAG1.M9TEMP1.M5LIST -----				LINE 00000055 COL 001 020			
COMMAND ==>				SCROLL ==> PAGE			
RUN	STMT	FILE	FILE	SYNCHRONIZATION		SYNC	SYNC
NAME	TYPE	NAME	TYPE	FIELD 1	FIELD 2	QUAL	METH
(EMUPDATE)(RF,EMPFILEX )(FILE )				(U)			
**	MK5WC03	TYPE 0	SCREEN DEFINITION EMMENU	CATALOGED	PRIOR	TO	VALIDATION.
**	MK5WC03	TYPE 0	SCREEN DEFINITION EMBROWSE	CATALOGED	PRIOR	TO	VALIDATION.
**	MK5CW03	TYPE 0	SCREEN DEFINITION QUIT	CATALOGED	PRIOR	TO	VALIDATION.
**	MK5CP01	TYPE 0	PREPROCESSING INPUT STREAM REQUEST REQ1				
STMT	FIELD	FIELD	FLD	DEC	OUTPT	EDIT	INITIAL
TYPE	NAME	LNGLTH	TYP	PLC	EDIT	LGTH	VALUE
COMMENTS							
(TF) (MYBLANK ) ( 10) (C)							
**	MK5RT12	TYPE 3	FILE DEFINITION FILE EMPFILEX	NOT FOUND ON LIBRARY.			
**	MK5CZ03	TYPE 5	RUN TERMINATED DUE TO ABOVE ERRORS.				
**	MK5W204	TYPE 0	NUMBER OF MESSAGES PRINTED IS 6.				
			M5INPUT	-	63	RECORDS	INPUT
			M5PUNCH	-	3	RECORDS	OUTPUT
			M5MAP	-	0	RECORDS	OUTPUT
			M5TPTBL	-	0	RECORDS	OUTPUT
			M5LIST	-	86	RECORDS	OUTPUT
			10	TRACKS	ASSIGNED	TO	M5LIB - 7 TRACKS NOT FULL.
			3	TRACKS	ASSIGNED	TO	M5LIB1 - 1 TRACKS NOT FULL.
**	MK5W209	TYPE 0	2176 BYTES OF TYPE A STORAGE USED DURING APPLICATION GENE				
***** BOTTOM OF DATA *****							

Figure 2-83 Browsing VISION:Transact Validation Results

When you are done browsing this data set, VISION:Workbench for ISPF automatically deletes it.

If no errors are found during the validation, the APPGEN listing is not displayed. Processing automatically proceeds to the Save panel.

## DB2 Considerations

If your VISION:Transact application contains any DB2 files, a DB2 environment is needed to successfully complete the validation. For this release, VISION:Workbench for ISPF is unable to dynamically invoke a DB2 environment for the purpose of completing a global validation. As a result, DO NOT request a global validation for VISION:Transact applications that contain DB2 files unless a DB2 environment is already established prior to invoking the global validation.

If you attempt to validate a VISION:Transact application that contains DB2 files without the proper DB2 environment already being established, your VISION:Workbench for ISPF session will ABEND.

## Catalog Considerations

Global validation of your VISION:Transact application is accomplished by invoking an actual VISION:Transact APPGEN run. During this validation, all instream file and screen definitions are automatically cataloged according to their specified catalog option. For this reason, concatenate your validation library specifications, placing a test library in front of your production library, as shown in [Figure 2-84](#).

```
VALAPP5 ----- SPECIFY TRIAL COMPILATION LIBRARIES -----
COMMAND ==>

Specify the Comlibs that should be used to perform a Trial Compilation:

COMLIB ==> 'TEST.COMLIB'
COMLIB1 ==> 'PROD.COMLIB'
COMLIB2 ==>
COMLIB3 ==>

Trial Compilation Notes:
- Catalog actions specified for instream Screen Definitions and
  Requests will be performed as part of the Trial Compilation.

- If this application contains DB2 File Definitions, enter 'END' to
  bypass validation.

Press ENTER to Perform a Trial Compilation
Enter END to By-Pass Trial Compilation and proceed to SaveAsis option
Enter CANCEL to Terminate Edit processing Without Saving the current item
```

Figure 2-84 Concatenating Validation Libraries

Use a Catalog option of R (for Replace) on all instream definitions so catalog errors are not encountered during the validation process.

When you are satisfied that your application is complete and you are ready to replace all instream definitions back into the production library, you can remove the test COMLIB from your validation library specifications. This replaces the definitions back into the production COMLIB.

## Using the Screen Developer

VISION:Workbench for ISPF with VISION:Transact provides interactive screen painting using the VISION:Workbench for ISPF Screen Developer facility. You can use the Screen Developer facility to create stand-alone screens by selecting Screen (option 23) from the Main menu.

Because run control information is automatically generated when stand-alone screens are exported by VISION:Workbench for ISPF, these screens are ready to submit and can be cataloged using the VISION:Transact Generate option in the VISION:Workbench for ISPF Main menu.

The VISION:Workbench for ISPF Screen Developer subsystem can also be used to create instream screens within a VISION:Transact application. If you request a global validation of a VISION:Transact application, all instream definitions, including screens, are cataloged during the validation according to the catalog action specified for each definition. If you do not request a global validation, you can use the VISION:Transact generate option to submit the application. This also catalogs your instream definitions according to the catalog option specified for each definition.

## Completing Screen Developer Options

Once you have entered the Screen Developer subsystem, the first panel you see is the Screen Developer Menu panel.

The following discussion on using the Screen Developer is based on creating the VISION:Transact screen shown in [Figure 2-85](#).

Customer Order Summary		
Customer Number: _____		
Month: _____		
Order Summary		
Order Number	Total Sale Amount	Status
_____	_____	_____
_____	_____	_____
_____	_____	_____
_____	_____	_____
_____	_____	_____
_____		

Figure 2-85 Sample Screen with Multiply-Occurring Screen Segments

To create a VISION:Transact screen, you select and complete each option shown on the Screen Developer Menu panel. Select and complete these options in the following order:

<u>PROFILE</u> (Option 0)	Specify layout parameters and define display attribute characters
<u>PROLOGUE</u> (Option 1)	Document the screen by providing comments
<u>PROCATTR</u> (Option 2)	Specify screen processing parameters
<u>LAYOUT</u> (Option 3)	“Paint” the screen
<u>SEGMENTS</u> (Option 6)	Define screen segments
<u>FIELDS</u> (Option 7)	Define screen data fields
<u>DISPATTR</u> (Option 4)	Assign display attributes to both data fields and literals
<u>VIEW</u> (Option 5)	Simulate a display of the completed VISION:Transact screen.

It is important that you follow this order. The DISPATTR and VIEW options function properly when the LAYOUT, SEGMENT and FIELD options are completed first.

### Using the PROFILE Option

Selecting PROFILE (option 0) from the Screen Developer Menu panel displays the Profile Option panel shown in [Figure 2-86](#).

SCRNPROF ---- MARKISPF .ISPLAG1.SOURCE(MENU) -----					
COMMAND ==>			SCROLL ==> CSR		
Field Designator ==>			(Default Field Designator - _ )		
Highlight Designator ==>			(Default Highlight Designator - @ )		
Device Identifier ==> T42132			(Default Device ID: T3270M2 )		
Line Cmd	Attribute Character	Bright/ Dark	Color	Extended HiLite	Program Symbol
0001	Z	H	N	-	-
0002	H	H	-	-	-
0003	B	H	B	-	-
0004	b	-	B	-	-
0005	D	D	-	-	-
0006	G	H	G	-	-
0007	g	-	G	-	-
0008	k	H	-	B	-
0009	K	-	-	B	-
0010	N	-	N	-	-
0011	P	H	P	-	-
0012	p	-	P	-	-
0013	R	H	R	-	-
0014	r	-	R	-	-
0015	T	H	T	-	-

Figure 2-86 The Profile Option Panel

The information on this Profile Option panel can be divided into two sections:

- Layout information.
- Attribute information.

The layout section of this panel provides VISION:Workbench for ISPF with the information necessary to process and generate your VISION:Transact screen layout source statements.

The layout information section contains the following entries:

*Field Designator* identifies what character is to be used to indicate the placement and length of data fields on the screen in the LAYOUT option. You do not have to make an entry in this field unless you want to override the default field designator character that is shown.

*Highlight Designator* identifies what character VISION:Workbench for ISPF should use to generate highlighted literals on VISION:Transact screen layout source statements. If you do not specify a highlight character, the default highlight character shown is automatically used by VISION:Workbench for ISPF when generating your screen layout source statements.

Unlike coding in native VISION:Transact, when using the Screen Developer to create screens, you do not use the highlight character within the LAYOUT option to indicate highlighting of literals. All display attribute characteristics are assigned through the Display Attribute option, DISPATTR.

*Device Identifier* identifies what type of screen you are generating. You can specify any device identifier that has been previously defined in your VISION:Transact device table.

As shown in [Figure 2-87](#), the VISION:Workbench for ISPF Parameters subsystem provides an option for browsing your site's VISION:Transact terminal device table.

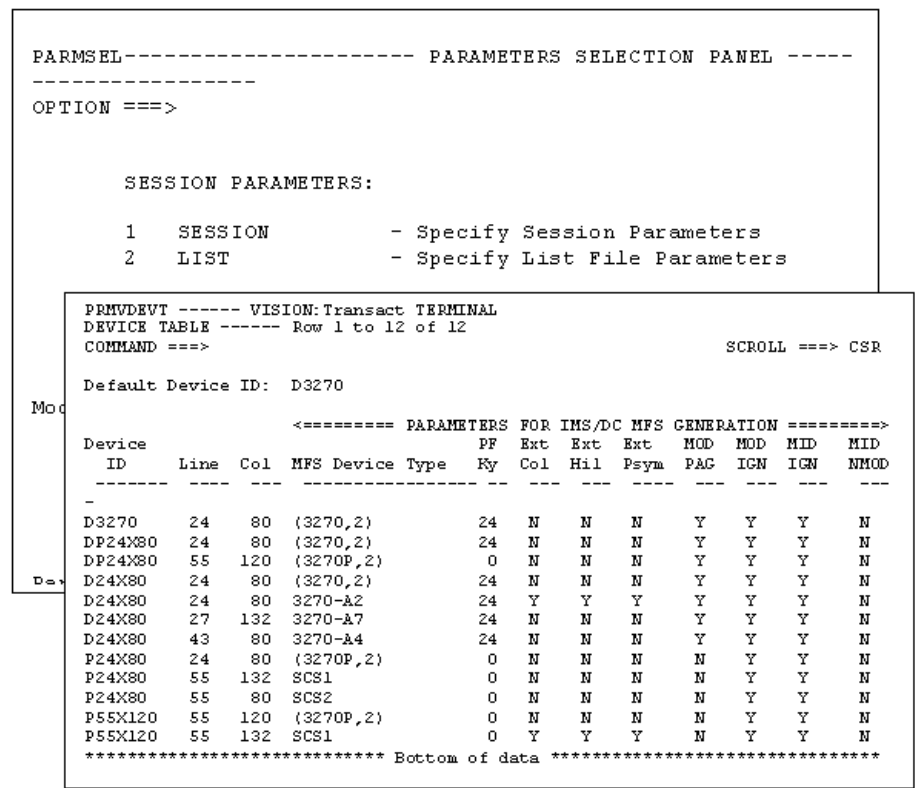


Figure 2-87 Browsing the Terminal Device Table

The attribute section of the Profile Option panel is used to define display attribute characters. These attribute characters can be used in the Display Attribute option, DISPATTR, to assign display characteristics to both literals and data fields.

Using the PROFILE option you can define 52 different display attribute characters. Valid characters are A - Z and a - z.

The first time that you use the Screen Developer, VISION:Workbench for ISPF automatically creates a set of default display attribute characters for you. This set of characters is shown in [Figure 2-88](#).

SCRNPROF ---- MARKISPF.ISPLAG1.SOURCE(MENU) -----					
COMMAND ==>			SCROLL ==> CSR		
Field Designator		==>		(Default Field Designator - _ )	
Highlight Designator		==>		(Default Highlight Designator - @ )	
Device Identifier		==> T42132		(Default Device ID: T3270M2 )	
Line Cmd	Attribute Character	Bright/ Dark	Color	Extended HiLite	Program Symbol
0001	Z	H	N	-	-
0002	H	H	-	-	-
0003	B	H	B	-	-
0004	b	-	B	-	-
0005	D	D	-	-	-
0006	G	H	G	-	-
0007	g	-	G	-	-
0008	k	H	-	B	-
0009	K	-	-	B	-
0010	N	-	N	-	-
0011	P	H	P	-	-
0012	p	-	P	-	-
0013	R	H	R	-	-
0014	r	-	R	-	-
0015	T	H	T	-	-

Figure 2-88 Screen Developer Default Attribute Set

You can modify this default set of attribute characters as needed. Any changes that you make are saved in your user profile so that whenever you use the Screen Developer you always have the same set of defined attribute characters.

If you import an existing screen that has a combination of display characteristics that have not been previously defined to a display attribute character, VISION:Workbench for ISPF automatically creates a new attribute character for you.

Using the PROLOGUE option

Selecting PROLOGUE (option 1) from the Screen Developer Menu panel, allows you to document your application by entering comments in the Prologue panel, shown in [Figure 2-89](#).

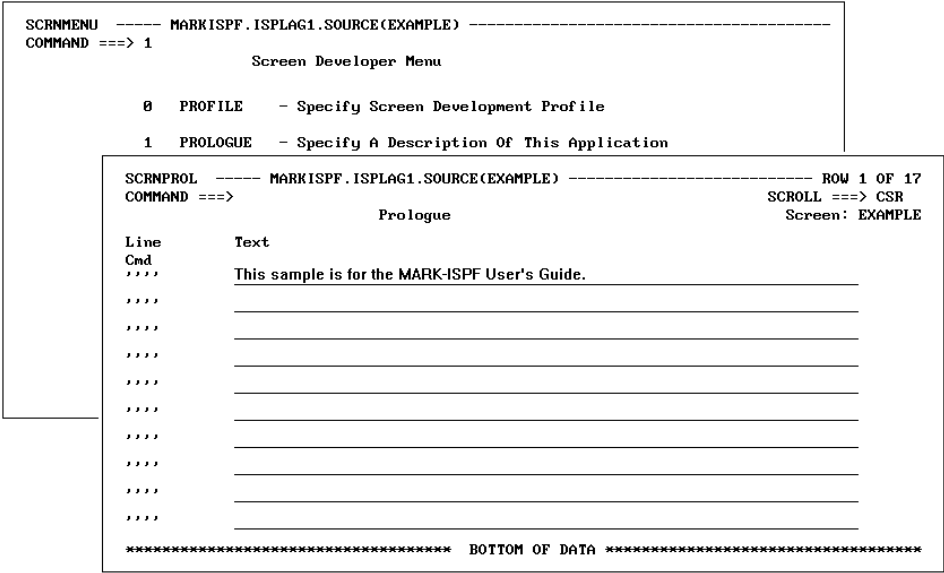


Figure 2-89 Adding Comments to Your Screen Definition

These comments become part of your screen definition source.



## Using the PROCATTR Option

Selecting PROCATTR (option 2) from the Screen Developer Menu panel displays the Processing Attribute panel. The PROCATTR option specifies VISION:Transact processing parameters for the screen you are creating. The Processing Attribute panel is shown in [Figure 2-90](#).

The image shows two overlapping terminal-style windows. The top window is titled 'SCRNMENU' and 'MARKISPF.ISPLAG1.SOURCE(EMUPDATE)'. It lists three options: 0 PROFILE, 1 PROLOGUE, and 2 PROCATTR. The bottom window is titled 'SCRNPATI' and 'MARKISPF.ISPLAG1.SOURCE(EXAMPLE)'. It contains various input fields for processing attributes, with 'Catalog Action' set to 'R'.

```

SCRNMENU  ----- MARKISPF.ISPLAG1.SOURCE(EMUPDATE) -----
COMMAND ==> 2
                                Screen Developer Menu                                Screen: EMMENU

      0  PROFILE   - Specify Screen Development Profile
      1  PROLOGUE  - Specify A Description Of This Application
      2  PROCATTR  - Specify Processing Attributes For This Screen

SCRNPATI  ----- MARKISPF.ISPLAG1.SOURCE(EXAMPLE) -----
COMMAND ==>
                                Processing Attributes                                SCROLL ==> CSR
                                                Screen: EXAMPLE

TARGET MONITOR                      ==>
BMS MAPSET Name                     ==>
Input to Transaction                 ==>
Process Empty Screen Segments:
  In TYPE 1 Requests?                ==>
                                AND After TYPE 1 Requests? ==>

Generate Multiple Output Pages? ==>

Catalog Action                      ==> R
Expiration date ==> __/__/__
                                User Data ==>

Suppress Printing of:
  Field Glossary?                   ==>
  Screen Layout?                    ==>
  
```

Figure 2-90 Specifying Screen Processing Attributes

Specify a catalog option of R (for Replace) on this panel.

## Using the LAYOUT Option

Selecting LAYOUT (option 3) from the Screen Developer Menu panel is where you actually “paint” your VISION:Transact screen. When painting your screen, literals are entered exactly as you want them to appear on your final VISION:Transact screen. The LAYOUT option does not automatically translate your entries to upper-case. The placement and length of data fields is indicated by using the field designator character that is specified in the PROFILE option.

Figure 2-91 shows a completed VISION:Transact screen layout. Notice that line numbers, along with the top two header lines, are provided for your convenience; they are not part of the screen definition.

```
SCRNMENU --- MARKISPF.ISPLAG1.M5INPUT(EXAMPLE) -----
OPTION ==> 3

      0 PROFILE   - Specify Screen Development Profile
      1 PROLOGUE  - Specify Descriptive Commetary For This Screen
      2 PROCATTR  - Specify Processing Attributes For This Screen
      3 LAYOUT    - Edit Screen Layout

SCRNLAYO --- MARKISPF.ISPLAG1.M5INPUT(EXAMPLE) -----
COMMAND ==>
                                                    SCROLL ==> CSR
                                                    SCREEN: EXAMPLE

0001                                     Customer Order Summary
0002
0003 Customer Number: _____
0004
0005 Month: _____
0006
0007                                     Order Summary
0008
0009
0010 Order Number      Total Sale Amount      Status
0011 _____
0012 _____
0013 _____
0014 _____
0015 _____
0016 _____
0017 _____
0019 _____
0020 _____
0021 _____
```

Figure 2-91 A Completed Screen Layout

If you are painting a screen that is too wide and/or long to view on the Layout Option panel at one time, you can use the standard ISPF primary scroll commands (RIGHT, LEFT, UP, and DOWN) to reposition your display so you can see the other areas of the screen that you are creating.

From the terminal device entry specified in the PROFILE option, VISION:Workbench for ISPF determines the size of the screen you are designing and lets you scroll accordingly.

You cannot scroll past the width boundaries of the screen as defined in your VISION:Transact terminal device table and you cannot save a screen that has more lines than allowed for the specified device.

For example, if you are creating a screen that is to be used on a 24 x 132 terminal (24 lines, 132 columns), VISION:Workbench for ISPF only lets you scroll to the right to a boundary of 132. If, while you are designing the screen you inadvertently add extra lines to the screen, VISION:Workbench for ISPF reminds you to delete the extra lines from the screen before allowing you to save it.

Unlike native VISION:Transact, in VISION:Workbench for ISPF, the LAYOUT option is not used to specify display attributes such as highlighting or color. All display attributes are assigned to literals and data fields in the Display Attribute option, DISPATTR. Attempting to use the VISION:Transact highlight designator within the LAYOUT option results in an error message.

## Using the SEGMENTS Option

Selecting SEGMENTS (option 6) from the Screen Developer Menu panel displays the Segments Option panel shown in [Figure 2-92](#).

```

SCRNMENU --- MARKISPF.ISPLAG1.M5INPUT(EXAMPLE) -----
OPTION ==> 6

      0 PROFILE      - Specify Screen Development Profile
      1 PROLOGUE     - Specify Descriptive Commentary For This Screen
      2 PROCATTR     - Specify Processing Attributes For This Screen
      3 LAYOUT       - Edit Screen Layout
      4 DISPATTR     - Specify Display Attributes For This Screen
      5 VIEW         - View Screen Layout
      6 SEGMENTS     - Specify Screen Segments
  
```

---

```

SCRNSEGS --- MARKISPF.ISPLAG1.M5INPUT(EXAMPLE) -----
COMMAND ==>
                                                    SCROLL ==> CSR
  
```

Line Cmd	Segment Number	Number Of Occurrences	Segment Level	Number Of Fields In Segment		Total
				Virtual	Non-Virtual	
0001	1	1	1	0	3	3
0002	2	5	2	0	3	3
0003	—	—				
0004	—	—				
0005	—	—				
0006	—	—				
0007	—	—				
0008	—	—				
0009	—	—				

\*\*\*\*\* END OF DATA \*\*\*\*\*

Figure 2-92 The Segments Option Panel

To complete the Segments Option panel, you make an entry for each segment that is contained within the screen you are creating. You can specify up to nine different segments. For multiply-occurring segments, it is important that you accurately complete the occurrence information as well. VISION:Workbench for ISPF cannot properly process or export screens containing multiply-occurring segments if the SEGMENTS option is not accurately completed.

Notice that there are Field Count entries on the right of each segment row. These are protected fields. VISION:Workbench for ISPF automatically calculates the number of fields that have been defined for each segment by reviewing the information that has been specified in the FIELDS option. Once the Fields option has been completed, these entries reflect the number and type of defined fields contained within each screen segment.

If you complete the FIELDS option before selecting the SEGMENTS option, default segment entries are automatically added to the Segments Option panel. These default entries are created by VISION:Workbench for ISPF from the information that you provided in the FIELDS option.

For every segment number referenced in the FIELDS option, VISION:Workbench for ISPF automatically adds a default segment entry to the Segments Option panel. As a result, if you want to delete an entry from the SEGMENTS Option panel you must first remove all references to that segment number from the FIELDS Option panel.

### Using the FIELDS Option

Selecting FIELDS (option 7) from the Screen Developer Menu panel allows you to define the data fields that have been specified with the placement of field designator characters in the LAYOUT option.

The FIELDS Option panel is shown in [Figure 2-93](#).

```

SCRNMENU --- MARKISPF.ISPLAG1.M5INPUT(EXAMPLE) -----
OPTION ==> 7

      0 PROFILE      - Specify Screen Development Profile
      1 PROLOGUE     - Specify Descriptive Commentary For This Screen
      2 PROCATTR     - Specify Processing Attributes For This Screen
      3 LAYOUT       - Edit Screen Layout
      4 DISPATTR     - Specify Display Attributes For This Screen
      5 VIEW         - View Screen Layout
      6 SEGMENTS     - Specify Screen Segments
      7 FIELDS       - Specify Screen Fields
  
```

---

```

SCRNFLDS --- MARKISPF.ISPLAG1.M5INPUT(EXAMPLE) -----
COMMAND ==>
                                                    SCROLL ==> CSR

                        Customer Order Summary

Customer Number: _____
Month: _____

                        Order Summary
=====
Line  In/  Screen  Q Dest  A  Q Source  Screen  Virtual  Layout
Cnd  Out  Field   L Field C  L Field  Segment Field   Position
      Name F Name  T  F Name  Number  ? Len   Line  Column
0001  B   CUSTNO  - CUSTNO M  - CUSTNO  01     -    3    20
0002  B   MONTH  - MONTH  M  - MONTH  01     -    5    10
0003  0   ORDERNO - - - - - ORDERNO 02     -   12    3
0004  0   AMOUNT  - - - - - AMOUNT  02     -   12   26
0005  0   STATUS  - - - - - STATUS  02     -   12   51
0006  0   MSGLINE - - - - - MSGLINE 01     -  999  999
0007  -   - - - - - - - - - - - 1     -  999  999
0008  -   - - - - - - - - - - - 1     -  999  999
  
```

Figure 2-93 The Fields Option Panel

Note that the Fields Option panel has two separate display areas. The top display area allows you to browse your screen layout as it is currently defined. It can be thought of as a miniature VIEW option. This can serve as a useful reminder when defining your screen data fields.

The bottom display area of this panel is where you actually start to define your data fields.

Use the following guidelines when completing the Screen Developer FIELDS option:

- Fields should be defined in the exact sequence (left to right, top to bottom) that they appear on the Layout Option panel.
- Only the first occurrence of multiply-occurring fields should be defined.
- With the exception of virtual fields, all fields must be assigned a unique screen field name. Duplicate screen field names must not be specified. Input/output fields should be specified as one entry, using the IN/OUT specification of B (for both).
- It is important that you accurately complete the Segment entry. All segment numbers entered here should have been previously defined in the SEGMENTS option. If you do not complete the SEGMENTS option first, default segment entries are added to the Segments Option panel for all segment numbers that are referenced on the Fields Option panel. As a result, to delete a default Segment entry from the SEGMENTS option, you must first remove all references of that segment number from the Fields Option panel.
- The Layout Position fields are informational entries that are automatically calculated by VISION:Workbench for ISPF. These positions represent the starting location of each data field that has been defined in the LAYOUT option. For multiply-occurring fields, only the starting location of the first occurrence is displayed.

In order for the layout position information to be correct, you must properly complete the LAYOUT and SEGMENT options prior to entering the FIELDS option. If you find that the specified layout positions do not correspond to the fields you are defining, incorrect data has been entered in either your LAYOUT, SEGMENTS, or FIELDS option.

[Figure 2-93](#), shows how preliminary field information for the sample screen is to be completed.

Notice how the fields are defined in the exact order that they appear on the Layout Option panel. Only the first occurrence of the multiply-occurring segment 2 fields are defined.

The bottom area of the Fields Option panel only allows you to specify preliminary information about each data field specified in the LAYOUT option. To continue entering additional information for a particular field, select that field with the Select line command as shown in [Figure 2-94](#).

```

SCRNFlds --- MARKISPF.ISPLAG1.M5INPUT(EXAMPLE) -----
COMMAND ==>
                                                                    SCROLL ==> CSR

                                Customer Order Summary

Customer Number: _____
Month: _____

                                Order Summary
=====
Line   In/   Screen   Q Dest   A   Q Source   Screen   Virtual   Layout
Cmd   Out   Field   L Field   C   L Field   Segment   Field     Position
      Name   F Name   T   F Name   Number   ? Len     Line   Column
0002  B     CUSTNO  _ CUSTNO  M   _ CUSTNO  01       _      3    20
0003  B     MONTH   _ MONTH   M   _ MONTH   01       _      5    10
0004  0     ORDERNO  _         _   _ ORDERNO  02       _     12     3
0005  0     AMOUNT   _         _   _ AMOUNT   02       _     12    26
0006  0     STATUS   _         _   _ STATUS   02       _     12    51
0007  0     MSGLINE  _         N   _ MSGLINE  01       _    999   999
0008  _         _         _         N   _         1         _    999   999
0008  _         _         _         N   _         1         _    999   999

```

Figure 2-94 Selecting a Data Field

The first follow-on Field Information panel is shown in [Figure 2-95](#).

```

SCRNBTHS--- MARKISPF.ISPLAG1.M5INPUT(EXAMPLE) ----- ROW 1 OF 1
COMMAND ==>
                                                                    SCROLL ==> CSR
                                                                    Screen : SCREEN
                                                                    Segment : 1
                                                                    Field : CUSTNO

In/   Numeric   Character
Out   Edit?     Edit Mask
B     _         _

                                Source Information
-----
Q,Name   Part Fld
_ CUSTNO St Ln   Comment
-----

                                Destination Information
-----

InpReq? ==> _   NumLok? ==> _   DecVal? ==> _   FltVal? ==> _
Protct? ==> _   Cursor? ==> _   ErrHil? ==> Y   ShoMod? ==> _

Line   Part   Default Value   Part
Cmd   Q,Name   St Ln   Q,Name   St Ln
'''   _ CUSTNO   M         _         _
***** BOTTOM OF DATA *****

```

Figure 2-95 The First Follow-On Field Information Panel

The Field Information panel is divided into three different sections:

- Screen field information.
- Source field information.
- Destination field information.

If you are defining an output field, you can use the source field section to *complete* your source field specifications. If the field you are defining is defined as an input-only field, do not make any entries in the source field section of this panel.

- Enter **SELECT** on the command line.
- Place the cursor on the source comment field.
- Press the Enter key.

Additional source field comments are entered in the Source Field Comments panel shown in [Figure 2-96](#).

[illegible]

If you are defining an input field, you can use the destination field section of this panel to *continue* your destination field specifications. If you are defining an output only field, you cannot complete any entries in the destination field section of this panel.

To continue processing any of these destination fields, use the **Select line** command to invoke an additional destination field panel as shown in [Figure 2-97](#).

```

SCRNBTHS --- MARKISPF.ISPLAG1.M5INPUT(EXAMPLE)----- ROW 1 OF 1
COMMAND ===> SCROLL ==> CSR
Screen : EXAMPLE
Segment : 01

In/ Numeric Character Segment : 01
Out
B SCRNDST --- MARKISPF.ISPLAG1.M5INPUT(EXAMPLE)----- Screen : SCREEN
COMMAND ==> Segment : 1
Field : CUSTNO

Part
Q,Name Act St Ln Comment
- CUSTNO M - - -

Message Override Code ==> _____ (Enter Text On Next Line)
_____

InpReq ---Dynamic Action Selection--- Validation-----
t A Q Control R Q Field Value: L C R Q B-Operand Partial
C L Field E L Constant or U O E L Field Name or S L O
T F Name L F Flag Field U N L F Constant T N P

Line
Cmd
S'''
*****

```

Figure 2-97 Specifying Additional Destination Field Information

Using this panel, you can complete your destination field information.

Notice that on this panel the following fields have a fixed amount of space for entries:

- Comment field.
- Message override code and text fields.
- Dynamic action specification.
- Dynamic validation specification.

If you need additional room to complete any of these areas follow these steps:

- Enter **SELECT** on the command line.
- Place the cursor on any of the fields listed above.
- Press the Enter key.

This invokes a follow-on panel related to the selected area. You can add additional entries to the follow-on panel.



Using the DISPATTR Option

Selecting DISPATTR (option 4) from the Screen Developer Menu panel displays a Display Attribute panel where you assign specific display characteristics to the literals and data fields on your VISION:Transact screen.

Using the sample VISION:Transact screen show in [Figure 2-92](#), selecting the DISPATTR option displays the screen shown in [Figure 2-98](#).

SCRNMENU----- MARKISPF . ISPLAG1 . M5INPUT(EXAMPLE) -----  
OPTION ==> 4

SCRNDATR --- MARKISPF . ISPLAG1 . M5INPUT(EXAMPLE) -----  
COMMAND ==> SCROLL ==> CSR

Customer Order Summary

Customer Number: \_\_\_\_\_  
Month: \_\_\_\_\_

Order Summary

Order Number	Total Sale Amount	Status
_____	_____	_____
_____	_____	_____
_____	_____	_____
_____	_____	_____
_____	_____	_____
_____	_____	_____

Figure 2-98     Assigning Display Attributes Characteristics

To assign a set of predefined display attribute characteristics to a particular literal or data field, you simply enter the corresponding display attribute character (defined in option 0) in the entry directly under the first character of the literal or data field.



SCRNDATR --- MARKISPF . ISPLAG1 . M5INPUT(EXAMPLE) ---  
COMMAND ==>>SCROLL ==> CSR

Customer Order Summary

Customer Number: \_\_\_\_\_  
Month: \_\_\_\_\_

Order Summary

Order Number	Total Sale Amount	Status
_____	_____	<div>R</div> _____
_____	_____	_____
_____	_____	_____
_____	_____	_____
_____	_____	_____

SCRNDATR --- MARKISPF . ISPLAG1 . M5INPUT(EXAMPLE) ---  
COMMAND ==>>SCROLL ==> CSR

Customer Order Summary

Customer Number: \_\_\_\_\_  
Month: \_\_\_\_\_

Order Summary

Order Number	Total Sale Amount	Status
_____	_____	<div>R</div> _____
_____	_____	<div>R</div> _____
_____	_____	<div>R</div> _____
_____	_____	<div>R</div> _____

Figure 2-100 Repeating Attributes in Multiply-Occurring Fields

- All occurrences of multiply-occurring data fields need not be assigned the same color attribute, but they must be assigned the same highlight attribute.

For example, as shown above, the remaining occurrences of the “Status” data field have been automatically assigned a highlighted-red attribute. By assigning “B” to the second occurrence of the “Status” data field, you change its display characteristics to highlighted-blue instead of highlighted-red.

Because the first occurrence of the Status field is specified as a highlighted field, VISION:Workbench for ISPF does not allow you to specify an unhighlighted attribute for any of the remaining occurrences of that field. If you try to change the display characteristic of the second occurrence to “b” for unhighlighted-blue, VISION:Workbench for ISPF automatically changes it back to “B” for highlighted blue.

If while viewing your screen you do not like the current arrangement of fields or display characteristics, you can use the LAYOUT or DISPATTR primary command to temporarily jump to either of these options. From these options, you can modify the screen as needed. When the modifications are done, use the END command to return to the VIEW option. You can see the results of your modifications immediately.

```
COMMAND ==>                                SCROLL ==> CSR
                                         MMDDYY HH.MM.SS
                                         -----

MENU SCREEN * * * * * * * * * * * * * * * * * * * * * * ORDER ENTRY

ENTER CUSTOMER NUMBER ==> -----

CHOOSE ONE OF THE FOLLOWING:

        C ... CUSTOMER

        O ... ORDER MAINTENANCE


ENTER YOUR CHOICE =====> -
```

[Figure 2-103](#) shows how the SHOWFLDS OFF primary command can be used to remove the data fields from the View screen.

```
COMMAND ==>                                SCROLL ==> CSR  
                                           MMDDYY HH.MM.SS
```

MENU SCREEN \* \* \* \* \* ORDER ENTRY

ENTER CUSTOMER NUMBER ==>

CHOOSE ONE OF THE FOLLOWING:

C ... CUSTOMER MAINTENANCE

O ... ORDER MAINTENANCE

ENTER YOUR CHOICE =====>

## Using Special Screen Developer Primary Commands

The Screen Developer primary commands are divided up into the following categories:

- Jump primary commands.
- View primary commands.
- Zoom primary commands.

Each type of command is discussed in the following subsections.

### Using Jump Commands

A special jump facility is provided within the Screen Developer subsystem. By entering the name of any Screen Developer option on the command line of a screen, you can freeze the option that you are currently working on and jump to the option that you entered on the command line.

When you are ready to return to the option from which you invoked the jump, simply use the END command.

The VISION:Workbench for ISPF Screen Developer subsystem supports the following subsystem specific primary commands:

#### *PROF or PROFile Command*

The minimum abbreviation for this command is four characters. While in the Screen Developer subsystem, the PROF or PROFile command freezes the option that you are currently working in and causes control to jump to the Profile option. When you have completed changes in the Profile option, you can return to the option from where you requested the profile jump by entering the END command.

#### *PROL or PROLogue Command*

The minimum abbreviation for this command is four characters. While in the Screen Developer subsystem, the PROL or PROLogue command freezes the option that you are currently working in and causes control to jump to the Prologue option. When you have completed changes in the Prologue option, you can return to the option from where you requested the prologue jump by entering the END command.

#### *PROC or PROCattr Command*

The minimum abbreviation for this command is four characters. While in the Screen Developer subsystem, the PROC or PROCattr command freezes the option that you are currently working in and causes control to jump to the PROCATTR option. When you have completed changes in the PROCATTR option, you can return to the option where you requested the PROCATTR jump by entering the END command.

*LA or LAYOUT Command*

The minimum abbreviation for this command is two characters. While in the Screen Developer subsystem, the LA or LAYOUT command freezes the option that you are currently working in and causes control to jump to the Layout option. When you have completed changes in the LAYOUT option, you can return to the option where you requested the LA or LAYOUT jump by entering the END command.

*DI or DISATTR Command*

The minimum abbreviation for this command is two characters. While in the Screen Developer subsystem, the DI or DISATTR command freezes the option that you are currently working in and causes control to jump to the DISATTR option. When you have completed your changes in the DISATTR option, you can return to the option where you requested the DI or DISATTR jump by entering the END command.

*F or Fields Command*

The minimum abbreviation for this command is one character. While in the Screen Developer subsystem, the F or Fields command freezes the option that you are currently working in and causes control to jump to the Fields option. When you have completed changes in the FIELDS option, you can return to the option where you requested the F or Fields jump by entering the END command.

*SE or SEGMENTS Command*

The minimum abbreviation for this command is two characters. While in the Screen Developer subsystem, the SE or SEGMENTS command freezes the option that you are currently working in and causes control to jump to the Segments option. When you have completed changes in the SEGMENTS option, you can return to the option where you requested the SE or SEGMENTS jump by entering the END command.

*VI or VIEW Command*

The minimum abbreviation for this command is two characters. While in the Screen Developer subsystem, the VI or VIEW command freezes the option that you are currently working in and causes control to jump to the View option. When you have completed changes in the VIEW option, you can return to the option where you requested the VI or VIEW jump by entering the END command.

## Using VIEW Option Commands

*SHowflds ON/OFF Command*

The minimum abbreviation for this command is two characters. This command is only active in the Screen Developer, VIEW option. It is used to add or remove field designators from the screen display.

When using the VIEW option, you can specify whether or not you want data fields to be displayed. If you choose to show data fields, they are represented with the field designator character.

To show data fields on the display, enter the SHOWFIELDS ON command on the command line, as shown in [Figure 2-104](#).

```
COMMAND ==> SHOWFLDS ON
```

SCROLL ==> CSR

MMDDYY HH.MM.SS  
XXXXXX XXXXXX

MENU SCREEN \* \* \* \* \* ORDER ENTRY

ENTER CUSTOMER NUMBER ==> XXXX

CHOOSE ONE OF THE FOLLOWING:

C ... CUSTOMER

O ... ORDER MAINTENANCE

ENTER YOUR CHOICE =====> X

Figure 2-104 The SHOWFLDS ON Command

To remove data fields from the display, enter the `SHOWFIELDS OFF` command on the command line, as shown in [Figure 2-105](#).

```
COMMAND ==> SHOWFLDS OFF
```

SCROLL ==> CSR  
MMDDYY HH.MM.SS

MENU SCREEN \* \* \* \* \* ORDER ENTRY

ENTER CUSTOMER NUMBER ==>

CHOOSE ONE OF THE FOLLOWING:

C ... CUSTOMER MAINTENANCE

O ... ORDER MAINTENANCE

ENTER YOUR CHOICE =====>

Figure 2-105 The SHOWFLDS OFF Command



## Using Zoom Processing Commands

Within the FIELDS option, there are panels that have a fixed amount of room for data entry. If you need to enter additional information for any of these fields, perform the following steps:

- Enter SELECT on the command line.
- Place the cursor on the field that you want to continue processing.
- Press the Enter key.

A follow-on panel for making additional entries is displayed.

## ASL Procedure Validation

When you complete your procedure and enter the END command, a complete, interactive validation of the procedure takes place. If no errors are found during this validation, VISION:Workbench for ISPF processes the END command as usual.

However, if errors are found, control returns to the Procedure panel where you will find error messages inserted into your procedure. The error messages are inserted following the procedure statements that caused the errors. These error messages are recognized by the EMSG literal inserted into the line command field, as shown in [Figure 2-106](#).

```

ASL4 ----- MARKISPF.ISPLAG1.M5INPUT(C3TEST2) -----
COMMAND ==>
PROCEDURE CONTAINS ERRORS, REFER TO 1 LINES MARKED "EMSG"
SCROLL ==> HALF

      Procedure Type ==>          Reinitialize Temps? ==>

Line  Free Form Processing Statements
Cmd  -----
'''  TEMP1:  FIELD F 2   INIT 0
'''  COUNT:  FIELD F 4   INIT 0 HEAD 'NO. OF'  'EMPLOYEES'
'''  TOTAL:  FIELD P 8 2 INIT 0 HEAD 'TOTAL'  'SALARY'
'''  AVG:    FEELD P 8 2 INIT 0 HEAD 'AVERAGE' 'SALARY'
EMSG M020 Invalid statement
'''  ;
'''  LET T.TEMP1 = 0 ;EXAMPLE OF LOOPING
'''  DO UNTIL T.TEMP1 = T.TEMP1 + 1
'''  LET T.TEMP1 = T.TEMP1 + 1
'''  END
'''  CALL REPORT REPORT1 ;PRINT REPORT1
'''  ;
'''  LET T.COUNT = 0
'''  LET T.COUNT = 0
'''  DO FORALL TEMPL UNTIL T.COUNT > 100
'''  LET T.COUNT = T.COUNT + 1 ;COUNT EMPLOYEES IN DEPARTMENT

```

Figure 2-106 Validating a Procedure

If you do not delete the error message lines before entering the END command again, rows that have the EMSG literal in the command line are automatically deleted when you exit the Procedure subsystem.



## The Unexpected Error Panel

When VISION:Workbench for ISPF experiences an error from which it cannot recover, the Unexpected Error panel is displayed. This screen is shown in [Figure 2-108](#).

```

ERROR ----- UNEXPECTED ERROR -----
COMMAND ==>
INVALID CURSOR FIELD/AREA SPECIFIED AS A PARAMETER

ID-DISPLAY SERVICE COULD NOT COMPLETE-RC > 8

*****
** WARNING: MARK-ISPF has encountered an Unexpected Error. **
**           It is possible that the function being performed did not **
**           complete. **
** **
** PROGRAM: INTEDISP 06/08/93 12.21 COMPUTER ASSOCIATES **
** **
** PSW : 00000000 00000000 **
** **
** REGISTERS AT TIME OF UNEXPECTED ERROR **
** **
** REGS 0 - 3 00000000 0005F8F0 0000000C 00000001 **
** REGS 4 - 7 0004F1CC 0004F1E4 00000000 0004F1F0 **
** REGS 8 - 11 0004F1BC 0005F4C4 0004EE48 00042000 **
** REGS 12 - 15 00048100 0005F880 4C0481CE 00000000 **
** **
** CALL COMPUTER ASSOCIATES INTERNATIONAL, INC. (CA) TECHNICAL SUPPORT. **
** **
*****
Enter END or RETURN

```

Figure 2-108 The Unexpected Error Panel

The Unexpected Error panel contains information that is very helpful to Technical Support when trying to track down the exact cause of a problem. If you should encounter this panel, please obtain a screen print when possible. If printing the screen is not possible, please copy the following information:

- Error Message.
- Primary Command.
- Pending Messages.
- PSW Content.
- Register 12 Contents.
- Register 14 Contents.
- Register 15 Contents.

In addition, please supply as much of the following information as possible:

- What part of VISION:Workbench for ISPF were you working in?
- Exactly what were you trying to do?
- What was the last data entry panel that you saw and/or completed?
- Any error messages that were displayed?
- Other comments that you feel might be helpful?

Once you have gathered as much of this information as possible, contact Computer Associates Technical Support.

## Contacting Computer Associates

For technical assistance with this product, contact Computer Associates Technical Support on the Internet at [esupport.ca.com](http://esupport.ca.com). Technical support is available 24 hours a day, 7 days a week.

# Index

## Symbols

---

? for help, 1-19, 2-18, 2-31

## A

---

action, 2-44, 2-49

allocate, 2-61, 2-64

APP4FIL Application Files panel, 2-103

APP4PARM Application Parameter panel  
  scan/terminate control entry, 2-102  
  source statement listing entry, 2-102

APPGEN application generation, 2-36, 2-103, 2-104

application items, 2-21

  applications, 2-21

  procedures, 2-21

  requests, 2-21

  screen definitions, 2-21

ASL, 1-9, 1-11, 2-32, 2-97

  procedure validation, 2-127

  Request and ASL Procedures panel, 2-92

Attention key, 1-7

## B

---

backup, 1-15, 1-16

batch, 1-1, 1-11

  application sample, 1-7

  execute applications, 1-11

  generate applications, 1-11

bookname, 2-45

browse, 2-70, 2-100

  default parameter modules, 2-70

  terminal device table, 2-108

  validation output, 2-32

buffer size, 2-43, 2-48, 2-55

## C

---

CALL statements, 2-128

catalog, 2-60, 2-111

  VISION:Transact considerations, 2-104

CICS/MVS, 1-7

CLIST, 2-60

  default, 2-60

  generate, 2-60

  pre-allocate ISPFIL, 2-83

COBOL Copybook Library Names panel, 2-45

COBOL definitions, 2-40

COBOL General Information panel, 2-41

COBOL Segment Information panel, 2-44

COMLIB, 2-23, 2-104

  backup, 1-16

---

- copy items, 1-17
- data set level enqueue (lock), 2-24
- edit, 2-20
- exclusive use, 2-24
- initialize, 1-15
- item name, 2-27
- maintenance, 2-68
- MARKLIBP parameter module, 2-70
- primary, 2-24
- utilities, 2-68
- validation, 2-30

COMLIB library name, 2-57

command stacking, 2-87

- attach item name, 2-88

commands, 2-4, 2-8

- line, 2-8, 2-13
- primary, 2-8
- zoom, 2-127

comments, 2-128

- additional destination field, 2-118
- additional source field, 2-117

concatenate libraries, 2-25

considerations

- COMLIB, 2-27
- DB2, 2-100
- partitioned data sets (PDS), 2-26
- sequential data set, 2-27

conversions

- input requests, 2-99
- output requests, 2-98

copy, 1-16, 2-29, 2-61

copybook library type, 2-52

creator, 2-50

## D

---

data entry, 2-4, 2-31

Data Set Specifications panels, 2-63

- batch, 2-63, 2-64
- foreground, 2-63, 2-65

data sets

- file tailoring output, 2-82
- list, 2-85
- partitioned (PDS), 2-26
- sequential, 2-27
- work, 2-100

DB2, 1-7, 2-100

- plan name, 2-74
- session parameters, 2-101
- subsystem ID, 2-74
- VISION:Transact considerations, 2-104
- VISION:Transact environment, 2-104

DB2 bind, 2-47

DB2 General Information panel, 2-47

DB2 plan name, 2-47

DB2 Segment Information panel, 2-49

DB2 subsystem ID, 2-47

DB2 table definitions, 2-46

DB2 table name, 2-50

defaults, 2-76

- display attribute characters, 2-77, 2-109
- Item Entry panel, 2-76
- job submission information, 2-77
- library, 2-88
- pad character, 2-72
- processing parameters, 2-86
- site parameters, 2-70
- Validation Library panel, 2-76

definition items, 2-21

- array definitions, 2-21
- file definitions, 2-21
- logical data views, 2-21
- table definitions, 2-21
- transaction definitions, 2-21

definition library, 2-41, 2-47, 2-51, 2-54, 2-56, 2-58

definitions

- data definitions, 1-2
- file definitions, 1-2
- instream, 2-96
- retrieve from COMLIB, 2-33

---

- device identifier, 2-107
- display attributes, 2-119
- duplicate entries, 2-32
- duplicate entry checking, 2-32
- duplicate field name check, 2-57
- duplicate field name editor, 2-59
- duplicate found, 2-59

## E

---

- edit, 2-21
  - application items, 2-21
  - definition items, 2-21
  - items, 2-31
  - select, 2-21, 2-22
  - source statement retrieval, 2-25
  - specify item location, 2-21, 2-22
  - specify source libraries, 2-23
  - specify validation libraries, 2-21
  - validate item, 2-21
- employee file (EMPFILE), 1-4, 1-8
- exporting source, 2-96

## F

---

- Facility
  - Jump, 2-124
- facility, 2-60
  - file tailoring, 2-63
  - help, 1-17, 2-1, 2-16
  - ISPF file tailoring, 2-60
  - ISPF help, 2-16
  - ISPF split screen, 2-20
  - submission, 2-60, 2-62, 2-77
  - tutorial, 2-16
- field definition name, 2-57
- field definitions, 1-6, 1-13
- field designator character, 2-107

- field display characteristics, 2-76
- field name prefix, 2-43, 2-48, 2-55
- field name validation, 2-31
- fields, 2-114, 2-115
  - additional destination field comments, 2-118
  - additional source field comments, 2-117
  - destination, 2-116
  - screen, 2-116
  - source, 2-116
- file definition name, 2-42, 2-48, 2-58
- file definitions, 1-2, 1-11, 1-13
  - create, 1-11
  - sample, 1-11
- file structures, 1-11, 1-13
- file tailoring, 2-60, 2-68
  - ISPF FILE, 2-61
  - output data set, 2-82, 2-84
  - skeleton, 2-60
- file type, 2-42
- file type Choice window, 2-42
- file types, 1-11, 1-12
- files
  - application, 1-9
  - Application Files panel, 2-103
  - master (old), 1-9
  - work, 1-9
- flags (flag names), 2-32
- from segment number/name, 2-59

## G

---

- GDBI, 2-102
- global validation, 2-30, 2-31, 2-74, 2-102
  - automatic, 2-33
  - VISION:Builder, 2-100
  - VISION:Transact, 2-103

---

## H

---

help, 1-1, 1-17, 2-16  
    ? in a field, 2-18  
    field level, 1-17  
    Help key, 1-18, 1-19, 2-88, 2-90, 2-91  
    HELP primary command, 1-18, 1-19, 2-88, 2-90, 2-91  
    panel level, 1-17  
    Summary message, 2-19  
highlight designator, 2-107

## I

---

Import External Definitions panel, 2-39  
import option  
    function keys, 2-40  
importing source, 2-96  
    guidelines, 2-96  
    VISION:Builder, 2-97  
    VISION:Transact, 2-97  
    VISION:Transact conversions, 2-97  
IMS, 1-7  
IMS/DB, 1-7  
initialize, 1-15  
    COMLIB, 1-15  
inquiry definition, 2-55  
invalid  
    duplicate entries, 2-32  
    flag names, 2-32  
ISPF, 1-2, 2-1  
    & variables in skeletons, 2-61  
    file tailoring, 2-60, 2-63, 2-68  
    jump, 2-91  
    LEFT & RIGHT scroll commands, 2-5  
    LIBRARY section, 2-24  
    multiple sessions, 2-20  
    primary commands, 2-9  
    profile library, 2-70

Split screen processing, 2-20  
split screen processing, 2-69  
UP & DOWN scroll commands, 2-5

ISPF, 2-61, 2-82  
    allocate/pre-allocate, 2-82  
    automatic allocate (VISION:Transact), 2-61, 2-83  
    file tailoring output data set, 2-82  
    manual allocate (VISION:Builder), 2-83  
    pre-allocate required, 2-61  
Item Entry panels, 2-23, 2-63  
    application, 2-23, 2-25  
    batch, 2-63  
    defaults, 2-76  
    definition, 2-23, 2-24  
    foreground, 2-63  
    ISPF, 2-64  
item names, 2-27  
    item name entry, 2-28  
    item selection list, 2-27  
    member name entry, 2-26, 2-28  
item selection list, 2-9, 2-27, 2-28  
items, 2-20  
    application items, 2-21  
    copy, 1-16  
    create, 2-29  
    default, 2-76  
    definition items, 2-21  
    edit, 2-21, 2-31  
    names, 2-28  
    save, 2-37  
    validate, 2-32

## J

---

JCL, 1-7, 2-60  
    automatically generate, 1-7  
    default, 2-60  
    generate, 2-60  
JCL/CLIST, 2-60  
    file tailoring, 2-61  
    generate, 2-60, 2-67



---

- Keep, 2-66
- submit, 2-66
- job submission, 2-60, 2-77
  - background, 2-81
  - background/batch, 2-79
  - file information, 2-64
  - foreground, 2-80, 2-82
  - processing options, 2-66
  - specify item, 2-63
  - submission facility, 2-62
  - user panel, 2-67
- jump, 2-91

## L

---

- level, 2-45, 2-49
- librarian library, 2-46, 2-52
- libraries
  - backup, 1-15, 1-16
  - COMLIB, 2-23
  - concatenate, 2-25, 2-104
  - copy items, 1-16
  - initialize, 1-15
  - ISPF, 2-23
  - partitioned or sequential data set, 2-23
  - profile, 2-70
  - restore, 1-15
  - sequential data set, 2-27
  - specify validation, 2-30
- line commands, 2-8, 2-13
  - ISPF, 2-14
  - pass, 2-31
  - repeat, 2-15, 2-16
  - select, 2-1, 2-14, 2-16, 2-28, 2-29, 2-118
  - using multiple, 2-94
  - VISION:Workbench, 2-14
- link screens, 1-6
- List data set
  - M9LIST, 2-85
- list data set, 2-69, 2-75, 2-85

- default processing parameters, 2-86
- dynamic allocation, 2-86
- holds output, 2-85
- Process List Dataset panel, 2-86
- set parameters, 2-75
- termination processing, 2-85

listing data set name, 2-41, 2-47, 2-51, 2-54, 2-57

## M

---

- M4LIST1, 2-102
  - source statement listing entry, 2-102
- M4OLD, 1-9
- M4PARAMS, 2-70
- M4REPn, 2-103
  - File usage entry, 2-103
- M4REPO, 1-9
- M9BGTS skeleton, 2-66
- M9BGUPNL panel, 2-66
- M9FGTS skeleton, 2-66
- M9FGUPNL user panel, 2-66
- M9GCTPU1 user panel, 2-66
- M9GCTPU2 user panel, 2-66
- M9GCTSBG skeleton, 2-66
- M9GCTSFG skeleton, 2-66
- M9LIST, 2-85
  - pre-allocate, 2-85
- MARKLIBP, 2-70
- member names, 2-26, 2-28
- member selection list, 2-9, 2-64
- menus
  - application development
    - APP4DEVL/APP5DEVL, 2-94
  - list of primary menus, 2-94
  - Main, 1-15, 2-5, 2-22, 2-69, 2-77, 2-83, 2-105
  - Parameters Selection, 2-71
  - Primary Option, 2-5, 2-22
  - Primary Option PRIMOPT4/PRIMOPT5, 2-22,

---

- 2-94
- PRIMOPT4/PRIMOPT5, 2-16
- Request REQ4MENU/REQ5MENU, 2-94
- Screen Developer, 2-94, 2-105
- Utility Selection, 1-16, 2-68

messages, 2-4, 2-83

- EMSG, 2-127
- error, 2-16, 2-129
- informational, 1-2, 2-31
- message line, 2-4
- summary, 2-19
- Unexpected Error panel, 2-129
- use FIND primary command, 2-33

MVS library, 2-46

MVS PDS library, 2-52

## N

---

- name, 2-45, 2-49
- name type, 2-59
- names validation, 2-30
- navigation, 2-87
  - jump facility, 2-91
- new definition name, 2-57
- number, 2-44, 2-49

## O

---

- online, 1-1
  - application sample, 1-3
  - execute applications, 1-7
  - generate applications, 1-7
- option numbers, 2-22, 2-87
- OPTIONS
  - PROLOGUE, 2-12
- options
  - application request, 2-103
  - background, 2-63, 2-81
  - background/batch, 2-79

- backup, 1-16
- catalog, 2-104
- default site parameters, 2-70
- DISPATTR, 2-11, 2-106, 2-107, 2-108, 2-119
- Display Attribute, 2-107, 2-108
- FIELDS, 2-11, 2-106, 2-113, 2-114, 2-115
- Foreground, 2-60, 2-63, 2-80, 2-82
- Generate, 2-63, 2-105
- initialize, 1-15
- Jumping, 2-91
- Keep, 2-65, 2-82
- Keep/Submit, 2-82
- LAYOUT, 2-106, 2-107, 2-111, 2-115
- Layout Option panel, 2-112
- List, 2-69, 2-71, 2-75
- Modifications, 2-69
- New Name, 2-29
- Options panel, 2-63, 2-77, 2-78, 2-83
- Parameters, 2-69, 2-71
- PD, 2-86
- Primary Option, 2-16
- PROCATTR, 2-12, 2-106, 2-111
- PROFILE, 2-12, 2-106, 2-108, 2-112
- Profile Option panel, 2-120
- PROLOGUE, 2-106, 2-110
- replace, 2-111
- screen, 2-105
- SEGMENTS, 2-12, 2-106, 2-113, 2-114, 2-115
- Segments Option panel, 2-115
- SELECT, 2-13
- session, 2-69, 2-71
- source statement retrieval, 2-25
- utilities, 1-15, 2-68
- VIEW, 2-13, 2-106, 2-115, 2-122, 2-123, 2-125

output

- application, 2-60
- list data set, 2-75
- M9LIST, 2-85
- printed, 2-69
- route to terminal, 2-65
- screens, 1-7
- validation, 2-32

---

## P

---

- pad characters, 2-72
- panel hierarchy, 2-1, 2-2, 2-31
- panel ID, 2-7
- panel names
  - suffix, 2-7
- panel sections, 2-4
  - fixed data entry area, 2-4
  - header, 2-4
  - scrollable data entry area, 2-4
- panels, 1-1, 1-3, 2-3, 2-22, 2-87, 2-89, 2-94
  - additional destination field, 2-118
  - Application Files, 2-92, 2-93, 2-103
  - application files, 1-4, 1-8
  - Application Item Entry, 2-23, 2-25
  - Application Parameter, 2-102
  - Batch Data Set Specifications, 2-64, 2-65
  - Batch Item Entry, 2-63
  - Batch Submission User, 2-67
  - column heading, 1-14
  - COMLIB backup, 1-16
  - COMLIB Initialization, 1-15
  - copy destination, 1-17
  - copy utility, 1-16
  - Data Set Specifications, 2-63, 2-64
  - Default Data Set Specifications, 2-64
  - Default user panel, 2-66
  - Definition Item Entry, 2-23, 2-24
  - Development/Administration Facility, 2-5
  - Display Attribute, 2-119
  - field column headings, 1-13
  - Field Definitions, 2-18
  - field definitions, 1-13
  - Field Information, 2-116
  - Fields Option, 2-114, 2-116
  - File Definition, 2-14, 2-16
  - file definition, 1-12, 1-13
  - File Definition (Segment name & number), 2-74
  - File Definition (Segment name), 2-73
  - File Item Entry, 2-25
  - file options, 1-12, 1-14
  - Follow-on, 2-116
  - Foreground Data Set Specifications, 2-65
  - Help, 2-17
  - ISPF Item Entry, 2-64
  - Item Entry, 2-23, 2-63, 2-76
  - Layout Option, 2-112, 2-115
  - Name, 2-5
  - Options, 2-63, 2-66, 2-77, 2-79, 2-80, 2-81, 2-82, 2-83
  - output specifications, 1-9, 1-10
  - Parameters Selection Menu, 2-69, 2-71
  - parent (previous), 2-5
  - Primary Option, 2-5, 2-22
  - Procedure, 2-127
  - Process List Dataset, 2-86
  - Processing Attributes, 2-111
  - Profile Option, 2-106, 2-108, 2-120
  - Prologue, 2-110
  - Report Output Specifications, 2-103
  - report title, 1-10
  - Request and ASL Procedures, 2-92, 2-93
  - Request Attribute, 2-99
  - Request Processing, 2-128
  - Save, 2-27, 2-35, 2-36, 2-38
  - Screen and Application Flow, 2-98
  - screen and application flow, 1-7
  - Screen Control Input Screen List, 2-99
  - Screen Developer Menu, 2-105, 2-106, 2-110, 2-111, 2-113, 2-114, 2-119, 2-122
  - screen field, 1-6
  - screen Layout, 1-4
  - Segments Option, 2-113, 2-114
  - Session Parameters, 2-72
  - Source Field Comments, 2-117
  - Specify "Save Asis" Library, 2-37
  - specify "To" COMLIB pane, 1-17
  - Specify List Data Set Defaults, 2-75
  - Specify Trial Compilation Libraries, 2-36, 2-104
  - Specify Validation Libraries, 2-30, 2-34, 2-76
  - Typical format, 2-4
  - Unexpected Error, 2-129
  - User, 2-63, 2-67, 2-77, 2-78, 2-79, 2-80, 2-81, 2-82

---

- utility Selection menu, 1-15
- Validation, 2-36, 2-76
- Panvalet library, 2-46, 2-52
- parameters, 2-75
  - list data set, 2-75
  - session, 2-69, 2-71, 2-72, 2-101
- Parameters Selection Menu panel
  - default site parameters group, 2-70
  - maintenance level group, 2-69
  - session parameters group, 2-69
- parameters subsystem, 2-69
- partitioned data sets (PDS), 2-26, 2-84
  - member names, 2-26
- PF keys, 1-3, 2-10
  - assignments, 2-10
  - KEYS primary command, 2-10
- primary commands, 2-4, 2-8, 2-9
  - CANCEL, 2-37
  - command stacking, 2-87
  - DISPATTR, 2-11, 2-122
  - DUP OFF, 2-11, 2-16
  - DUP ON, 2-11, 2-32
  - END, 1-19, 2-1, 2-12, 2-16
  - FIELDS, 2-11
  - HELP, 1-18, 1-19, 2-9, 2-16, 2-88, 2-90, 2-91
  - ISPF FIND, 2-33
  - ISPF LOCATE, 2-29
  - ISPF SCROLL DOWN, 2-32
  - ISPF SCROLL LEFT, 2-32
  - ISPF SCROLL RIGHT, 2-32
  - ISPF SCROLL UP, 2-32
  - KEYS, 2-10
  - LAYOUT, 2-122
  - Most ISPF commands active, 2-9
  - NAMES F, 2-11, 2-32
  - NAMES HELP, 2-11
  - NAMES OFF, 2-11, 2-31
  - NAMES ON, 2-11, 2-31
  - NAMES q, 2-11
  - NAMES STATUS, 2-11, 2-12
  - NOCHECK, 2-11, 2-12

- PANELID, 2-7
- PROCATTR, 2-12
- PROFILE, 2-12, 2-120
- PROLOGUE, 2-12
- SAVE, 2-38
- SAVE ASIS, 2-38
- screen developer, 2-124
- SEGMENTS, 2-12
- SELECT, 2-13, 2-28, 2-29
- SHOWFLDS OFF, 2-13, 2-123
- SHOWFLDS ON, 2-13, 2-123
- split screen, 2-69
- VIEW, 2-12, 2-13, 2-122
- primary/alternate field names, 2-59
- product menus, 2-22
- Profile Option panel
  - device identifier, 2-107
  - field designator, 2-107
  - highlight designator, 2-107
- profiles, 2-70
  - default information, 2-76
  - library, 2-70
  - PROFILE option, 2-106
  - save processing information, 2-77
  - screen developer profile, 2-76
  - screen display characteristics, 2-76
  - session profile information, 2-70
  - user, 2-70
  - user profile IADSPROF, 2-70, 2-76
- program function (PF) keys, 2-10

## R

---

- record size, 2-43
- records per block, 2-43
- rename the definition, 2-55
- reports, 1-7, 1-10
  - alternate report files, 2-103
  - define, 1-10
  - output specifications, 2-103

---

- samples, 2-101
- REQ4REPT Report Output Specifications panel, 2-103
  - Report Handling entry, 2-103
- requests, 2-21, 2-99
  - converted input, 2-99
  - converted output, 2-98
  - Request Attribute panel, 2-99
  - screen control statements, 2-98
  - stand-alone support, 2-99
- restore, 1-15
- results definition, 2-52
- run control statements, 2-27, 2-28, 2-97
- run types
  - APPGEN run, 2-32, 2-36, 2-103, 2-104
  - Scan run, 2-32, 2-100
  - Validation runs, 2-32

## S

---

- save, 2-35, 2-38
  - items, 2-37
- scan/terminate control entry
  - N for GDBI, 2-102
  - R for sample report, 2-102
- screen control statements, 2-98
- screen developer, 2-105
  - DI or DIspattr, 2-125
  - DISPATR, 2-106, 2-119
  - F or Fields, 2-125
  - FIELDS, 2-106
  - FIELDS option, 2-114
  - jump, 2-124
  - LA or LAYOUT, 2-125
  - LAYOUT, 2-106, 2-111
  - PROC or PROCattr, 2-124
  - PROCATR, 2-106, 2-111
  - PROF or PROFile, 2-124
  - PROFILE, 2-106, 2-112
  - PROL or PROlogue, 2-124

- PROLOGUE, 2-106, 2-110
- SE or SEgments, 2-125
- SEGMENTS, 2-106, 2-113
- SHowflds ON/OFF, 2-125
- special primary commands, 2-124
- VI or VIEw, 2-125
- VIEW, 2-106, 2-121, 2-122, 2-126
- zoom processing commands, 2-127

- screen layouts, 1-5
- screen painting, 2-76
- screens, 1-4, 2-98
  - create data entry screens, 1-4
  - define fields, 1-6
  - file maintenance, 1-5
  - initial menu, 1-5
  - input, 1-7
  - layouts, 1-4
  - limits, 2-112
  - link, 1-6, 1-7
  - paint, 1-2, 1-4, 1-6, 2-76
  - preview, 1-2
  - screen control statements, 2-98
  - split screen processing, 2-69
  - stand-alone, 2-105

- Segments
  - SEGMENTS option, 2-113
- segments, 1-13
  - multiply-occurring, 2-120
  - name, 2-73
  - number, 2-73
  - Segments Option panel, 2-113, 2-115
- select line command, 2-29
- SELECT primary command, 2-29
- session parameters, 2-9, 2-71, 2-72
  - autoskip, 2-72
  - DB2 environment, 2-100
  - DB2 PLAN NAME, 2-74
  - DB2 SUBSYSTEM ID, 2-74
  - delete character, 2-73
  - input pad character, 2-72
  - line command pad character, 2-72

---

- segment reference, 2-73
- specify parameters module, 2-74
- start locations, 2-74
- work area size, 2-74

session profile information, 2-70

source, 2-96

- exporting, 2-96
- guidelines, 2-96
- importing, 2-96
- retrieval, 2-25
- retrieve, 2-85, 2-96
- statement order, 2-97
- VISION:Builder, 2-97
- VISION:Transact, 2-97

split screen processing, 2-20, 2-69

- disabled, 2-20

start locations, 2-74

statement order, 2-97

structured panel hierarchy, 2-1

subsystems

- DB2, 2-74
- edit, 2-18, 2-64, 2-76
- file definition, 2-2, 2-14
- generate, 2-18
- parameters, 2-69, 2-71, 2-75, 2-86
- procedure, 2-12, 2-127
- screen developer, 2-12, 2-76, 2-77, 2-105, 2-124
- utilities, 2-20, 2-25, 2-68, 2-69, 2-73, 2-85

System Modifications (SMs), 2-69

## T

---

terminal device table, 2-108

titles

- panel title line, 2-4

trial compilation, 2-36

TSO, 1-2

- ALLOCATE command, 2-83

tutorial, 2-16

type, source library, 2-45

## U

---

unloaded inquiry system data base, 2-54

user panels, 2-78

user profile IADSPROF, 2-70

utilities, 1-14, 2-68

- special delete character, 2-73
- specify work area, 2-74

## V

---

validation, 2-30, 2-74

- definitions, 2-33
- entries, 1-2
- field names, 2-31
- items, 2-21, 2-32
- libraries, 2-21, 2-30
- names, 2-30
- output, 2-32
- specify validation libraries panel, 2-76
- turn off, 2-31
- VISION:Builder applications, 2-30, 2-35, 2-100
- VISION:Transact applications, 2-36, 2-103

VISION:Builder, 1-1

- background/batch processing, 2-63, 2-79
- create applications, 1-1, 1-7
- create file definitions, 1-1
- DB2 considerations, 2-100
- foreground processing, 2-80
- import source, 2-96, 2-97
- M4PARAMS parameter module, 2-70
- maintain applications, 2-96
- PRIMOPT4, 2-22, 2-71
- validation considerations, 2-100
- validation libraries, 2-30

VISION:Builder definitions, 2-56

VISION:Builder Information panel, 2-56

VISION:Inquiry definitions, 2-53

---

VISION:Inquiry Information panel, 2-54

VISION:Results definitions, 2-50

VISION:Results Information panel, 2-51

VISION:Transact, 1-1

- background processing, 2-81

- conversions, 2-97

- create application screens, 1-2

- create applications, 1-1, 1-4

- create file definitions, 1-1

- DB2 environment, 2-104

- FIVEPARAM parameter module, 2-70

- foreground processing, 2-82

- import source, 2-96, 2-97

- maintain applications, 2-96

- PRIMOPT5, 2-22, 2-71

- screen painting, 2-105

- screen preview, 1-2, 2-122

- terminal device table, 2-70, 2-108

- trial compilation, 2-36

- validation considerations, 2-103

VISION:Workbench

- create batch applications, 1-1

- create online applications, 1-1, 1-3

- help, 1-1, 1-2, 1-17, 2-16

- overview, 1-1

- panel names, 2-5

- primary commands, 2-32

- tutorial, 1-2, 2-16

- utilities, 1-14

VISION:Workbench for DOS, 2-96

VSAM, 1-7

## Z

---

zoom processing commands, 2-127

